



Universidad
Carlos III de Madrid

Ingeniería Técnica en Informática de Gestión

PROYECTO FIN DE CARRERA

Desarrollo de un sistema de diálogo oral para la provisión de información cinematográfica

Autor: José Antonio Aceves Llorente

Tutor: Dr. David Griol Barres

Leganés, octubre de 2015

Título: Desarrollo de un sistema de diálogo oral para la provisión de información cinematográfica

Autor: José Antonio Aceves Llorente

Director: Dr. David Griol Barres

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

En primer lugar, me gustaría dar las gracias a David, mi tutor, por su gran ayuda, dedicación y empuje que ha mostrado desde el primer día para que este proyecto salga adelante, sin ti David esto no hubiera sido posible. Muchas gracias.

Quiero dar las gracias también a toda mi familia por el apoyo incondicional que siempre ha mostrado. En especial, a mis tíos o mejor dicho, a “mis padres de Madrid”, Leonor y Maximino, por acogerme en su casa durante tantos años y tratarme y quererme como a un hijo más. Por supuesto, también quiero dar las gracias a mi primo Alberto, ya que durante muchos años le invadí su territorio y él nunca puso ni una pega sobre ello. Gracias a los 3, para mi sois más que mis tíos y mi primo.

Muchos años han sido los que he estado fuera de mi casa, alejado de mis padres, Ramona y José Antonio y de mis hermanos, Juan y Virginia, pero esa distancia no ha impedido que siempre hayan estado ahí cuando les he necesitado y apoyándome en todo. Muchas gracias a todos, me habéis ayudado mucho, me habéis dado muy buenos consejos y por supuesto no os habéis olvidado de mí en ningún momento.

Gracias a todos los compañeros que he tenido en la universidad, Edu, Alberto, Natalia, Víctor, David, Alfonso, Diego, que sin ese grupo tan magnífico que conseguimos formar no hubiéramos podido sacar la carrera adelante. Gracias a todos por ser como sois, no cambiéis.

Resumen

El objetivo principal de este Proyecto Final de Carrera es el desarrollo de un portal de voz de información de películas utilizando el lenguaje de programación VoiceXML. A través de este portal los usuarios podrán telefónicamente mediante voz o teclado para acceder a información sobre películas y sesiones de cine.

El portal de voz ofrece al usuario distintas funcionalidades: búsqueda de películas (búsquedas por título, director o actor), creación de un perfil con los gustos sobre cine del usuario (actores, directores y géneros preferidos y rechazados, horarios para ir al cine y cines favoritos), acceso a un servicio de recomendación de sesiones de cine basado en el perfil del usuario, información sobre películas en cartelera, información sobre próximos estrenos, acceso a información de películas (argumento, valoración de FilmAffinity, valoración en base a los gustos del usuario, director, reparto, productora, año, críticas de prensa, críticas de usuarios de FilmAffinity, etc.). Toda esta información es extraída de la página www.FilmAffinity.com.

Gracias al sistema desarrollado, el usuario podrá acceder a un servicio de información de películas y sesiones de cine adaptado a sus gustos que le recomendará qué ir a ver al cine y cuando de una forma rápida y sencilla.

La aplicación se apoya en una infraestructura digital MVC desarrollada también en este Proyecto Final de Carrera que permite crear páginas VXML de forma rápida y sencilla, permitiendo al desarrollador crear una aplicación VXML sin necesidad de tener que escribir VXML. Con esta infraestructura digital hace que la aplicación sea fácilmente extensible, dinámica, flexible y fácil de probar, aumentando de esta manera su calidad y eficiencia.

Tanto la aplicación como la infraestructura digital se apoyan en otras tecnologías, como son la utilización de bases de datos, servidores web, VoiceXML (Voxeo Evolution y Free Web Hosting Area) y uso de diferentes lenguajes de programación (SQL, PHP y HTML).

El proyecto se complementa con un estudio detallado y análisis de los sistemas de diálogo y de la aplicación del estándar VoiceXML para su desarrollo, llevado a cabo con el objetivo de aplicar todo su potencial para completar con éxito este Proyecto Final de Carrera.

Palabras clave: Sistemas de diálogo, FilmAffinity, portal de voz, VoiceXML, Voxeo.

Abstract

The main objective of this Bachelor Project is to develop a voice portal to provide films information using the VoiceXML programming language. By means of the developed portal, users can interact via voice or the telephone keypad to access to films and cinema shows information.

The developed voice portal offers citizens different functions: search films (Search by title, actor and director), create profile about films preferences (liked and disliked genres, actors and directors, favourite cinema schedule and favourite cinemas), access to a show recommender based on user's profile, information about current movie listings and next releases, detailed film information (story line, FilmAffinity rating, rating based on user's profile, director, cast, producer, year, press reviews, FilmAffinity user's reviews, etc.). All this information is extracted from www.filmaffinity.com.

Thanks to the developed application, users will be able to access to movie and cinema show information service adapted to their preferences. This service will be able to recommend what show to watch in which cinema and when is the best schedule quickly and easily.

The voice portal is developed on MVC Framework developed in this Bachelor Project. This Framework allows to create VXML pages quickly and easily and it allows to developer create a VXML application without the need of to use VXML Language. Which make it more dynamic, extensible, testable and flexible increasing its quality and efficiency.

The application and the Framework integrate other technologies, such as the use of databases (MySQL), web, VoiceXML servers (Voxeo Evolutions and Free Web Hosting Area) and several programming languages (SQL, PHP and HTML).

The project has been extended with a detailed study and analysis of spoken dialog systems and the application of the VoiceXML standard for their implementation. This study has been very useful to learn and apply the full potential of these technologies for the complete development of the voice portal.

Keywords: Spoken dialog systems, FilmAffinity, voice portal, VoiceXML, Voxeo.

Índice

CAPÍTULO 1. Introducción y Objetivos.....	17
1.1 Antecedentes	17
1.2 Objetivos	18
1.3 Fases de desarrollo	21
1.4 Planificación temporal	22
1.4.1 Desglose de tareas y Diagrama de Gantt	23
1.5 Medios y documentación empleados	24
1.6 Presupuesto	25
1.6.1 Recursos.....	25
1.6.2 Resumen de costes.....	26
1.7 Estructura de la memoria	27
CAPÍTULO 2. Estado del Arte.....	28
2.1 Introducción a los sistemas de diálogo	28
2.2 Evolución de los sistemas de diálogo	28
2.2.1 Primeros sistemas de diálogo (años 60).....	28
2.2.2 Procesamiento del lenguaje natural (Años 70)	29
2.2.3 Tecnologías del Habla	31
2.2.4 Retos actuales	34
2.3 Arquitectura de los sistemas de diálogo.....	35
2.4 Clasificación de los sistemas de diálogo.....	36
2.4.1 Iniciativa del diálogo	36
2.4.2 Dependencia del hablante.....	36
2.4.3 Tipo de comunicación	37
2.4.4 Tipo de idioma	37
2.4.5 Tipo de discurso.....	37
2.4.6 Adaptación	38
2.5 VoiceXML	38
2.5.1 Conceptos básicos de VoiceXML.....	39
2.5.2 Constructores de diálogo.....	42
2.5.3 Gramáticas.....	45
CAPÍTULO 3. Sistema desarrollado.....	47
3.1 Descripción de la infraestructura digital utilizada.	47
3.2 Lógica de navegación (Controladores)	47
3.2.1 NavigationMap	48
3.2.2 Controllers	49
3.3 Lógica de pintado (Vistas)	53
3.3.1 Templates	53
3.3.2 ViewData.....	54
3.3.3 Clases de Apoyo.....	65
3.4 Modelo (Apis y Sistemas de almacenamiento).....	66
3.4.1 Modelo de datos externo.....	66
3.4.2 Modelo de datos locales.....	73

3.4.3	Modelo de datos mixto	76
3.5	Diagrama de secuencia del sistema.....	78
CAPÍTULO 4. Descripción detallada del sistema.....		79
4.1	Menú Principal.....	79
4.1.1	Diálogos.....	80
4.2	Menú Buscar	80
4.3	Menú Cartelera.....	82
4.3.1	Diálogos.....	83
4.4	Menú Próximos Estrenos	84
4.4.1	Diálogos.....	85
4.5	Identificación	86
4.5.1	Diálogos.....	87
4.6	Registro	89
4.6.1	Diálogos.....	89
4.7	Perfil.....	91
4.7.1	Diálogos.....	92
4.8	Gestionar provincia.....	92
4.8.1	Diálogos.....	93
4.9	Gestionar Horario.....	94
4.9.1	Diálogos.....	95
4.10	Gestionar Directores	97
4.10.1	Diálogos.....	97
4.11	Gestionar Actores.....	99
4.11.1	Diálogos.....	99
4.12	Gestionar Cines.....	101
4.12.1	Diálogos.....	101
4.13	Que Ver (Sistema de recomendación de películas)	103
4.13.1	Sistema de recomendación de películas	103
4.13.2	Diálogos.....	104
4.14	Listado paginado de películas.....	106
4.14.1	Diálogos.....	107
CAPÍTULO 5. Evaluación de la aplicación.....		111
5.1	Pruebas automáticas.....	111
5.2	Pruebas manuales y evaluación	112
5.2.1	Prueba 1: Usuario sin identificar navega por la cartelera y por los próximos estrenos	113
5.2.2	Prueba 2: Usuario sin identificar busca una película por director, actor y por título.	116
5.2.3	Prueba 3: Usuario consulta una película y sus críticas.....	119
5.2.4	Prueba 4: Usuario sin registrar se registra y rellena su perfil.....	122
5.2.5	Prueba 5: Usuario identificado consulta las recomendaciones para ir a ver al cine	125
5.2.6	Conclusiones de las pruebas manuales	128
CAPÍTULO 6. Conclusiones y trabajo futuro		129
6.1	Conclusiones	129

6.2	Trabajo futuro.	130
6.2.1	<i>Trabajo futuro para mejorar la infraestructura digital.....</i>	<i>130</i>
6.2.2	<i>Trabajo futuro para mejorar FilmAffinityVxml.....</i>	<i>131</i>
GLOSARIO		133
BIBLIOGRAFÍA.....		135

Índice de Figuras

Figura 1-1 - Desglose de las tareas y planificación temporal del proyecto	23
Figura 1-2 - Diagrama de Gantt con la planificación temporal del proyecto.....	23
Figura 2-1 - Arquitectura de un sistema de diálogo.....	35
Figura 2-2 - Diagrama de interacción de un intérprete VXML con un servidor web.....	39
Figura 2-3 - Ejemplo de formulario directo	43
Figura 2-4 - Ejemplo de Menú VXML.....	44
Figura 3-1 - Array Navigation Map.....	48
Figura 3-2 - Diagrama de clases de Controllers	52
Figura 3-3 - Diagrama de clases de ViewData.....	63
Figura 3-4 - Diagrama de clases de Views.....	64
Figura 3-5 - Diagrama de secuencia del sistema	78
Figura 4-1 - Diagrama de flujo del Menú principal.....	79
Figura 4-3 - Diagrama de flujo Menú cartelera	83
Figura 4-4 - Diagrama de flujo Menú Próximos Estrenos	85
Figura 4-5 - Diagrama de flujo de identificación	87
Figura 4-6 - Diagrama de flujo del registro.....	89
Figura 4-7 - Diagrama de flujo Perfil.....	91
Figura 4-8 - Diagrama de flujo Gestionar Provincia	93
Figura 4-9 - Diagrama de flujo Gestionar horario	95
Figura 4-10 - Diagrama de flujo Gestionar Directores	97
Figura 4-11 - Diagrama de flujo Gestionar Actores	99
Figura 4-12 - Diagrama de flujo Gestionar Cines.....	101
Figura 4-13 - Diagrama de flujo Que ver.....	104
Figura 4-14 - Diagrama de flujo Listado de películas	106

Índice de Tablas

Tabla 1-1 - Detalle de Costes de Recursos Humanos del Proyecto	26
Tabla 1-2 - Detalle del Coste Total del Proyecto	26
Tabla 3-1 - Gramática generada por la ViewData.....	55
Tabla 3-2 - Ejemplo del CSV utilizado para reconocer nombres de directores	57
Tabla 3-3 - Ejemplo de gramática creada para la búsqueda de directores	58
Tabla 3-4 - Gramática autogenerada para el formulario de identificación	59
Tabla 3-5 - Información básica del usuario.....	73
Tabla 3-6 - Información de una sesión	74
Tabla 3-7 - Información sobre los teléfonos	74
Tabla 3-8 - Preferencias del usuario.....	75
Tabla 3-9 - Información temporal de una película.....	75
Tabla 4-1 - Diálogo Menú principal para usuarios anónimos.....	80
Tabla 4-2 - Diálogo Menú principal para usuarios identificados	80
Tabla 4-3 - Diálogo Menú Buscar	81
Tabla 4-4 - Diálogo Búsqueda por título	82
Tabla 4-5 - Diálogo Búsqueda por actor	82
Tabla 4-6 - Diálogo Búsqueda por director.....	82
Tabla 4-7 - Diálogo Menú Cartelera	83
Tabla 4-8 - Diálogo Cartelera por fecha de estreno.....	84
Tabla 4-9 - Diálogo Cartelera por puntuación.....	84
Tabla 4-10 - Diálogo Cartelera por popularidad.....	84
Tabla 4-11 - Diálogo Menú Próximos estrenos	85
Tabla 4-12 - Diálogo Próximos estrenos por fecha de estreno.....	86
Tabla 4-13 - Diálogo Próximos estrenos por puntuación.....	86
Tabla 4-14 - Diálogo Próximos estrenos por popularidad	86
Tabla 4-15 - Diálogo New Identificación.....	87
Tabla 4-16 - Diálogo Identificación por terminal.....	88
Tabla 4-17 - Diálogo Contraseña	88
Tabla 4-18 - Diálogo Identificación incorrecto	88
Tabla 4-19 - Diálogo Guardar terminal.....	88
Tabla 4-20 - Diálogo Registro	90
Tabla 4-21 - Diálogo Contraseña	90
Tabla 4-22 - Diálogo Repita Contraseña	90
Tabla 4-23 - Diálogo Error Usuario ya existe	90
Tabla 4-24 - Diálogo Error Contraseñas distintos.....	90
Tabla 4-25 - Diálogo Registro Ok	91
Tabla 4-26 - Diálogo Perfil	92
Tabla 4-27 - Diálogo Mostrar Provincia.....	93
Tabla 4-28 - Diálogo Sin Provincia.....	94
Tabla 4-29 - Diálogo Añadir Provincia.....	94
Tabla 4-30 - Diálogo Provincia guardada.....	94

Tabla 4-31 - Diálogo Mostrar Horario	95
Tabla 4-32 - Diálogo Sin Horario	96
Tabla 4-33 - Diálogo Añadir Hora inicio.....	96
Tabla 4-34 - Diálogo Añadir Hora Fin	96
Tabla 4-35 - Diálogo Error.....	96
Tabla 4-36 - Diálogo Horario guardado.....	96
Tabla 4-37 - Diálogo Gestionar Directores (Sin directores guardados)	97
Tabla 4-38 - Diálogo Gestionar Directores (Con directores guardados).....	98
Tabla 4-39 - Diálogo Añadir director (favorito o lista negra)	98
Tabla 4-40 - Diálogo Director favorito añadido	98
Tabla 4-41 - Diálogo Director añadido a lista negra	98
Tabla 4-42 - Diálogo Borrar Directores favoritos	98
Tabla 4-43 - Diálogo Lista negra de directores borrada.....	98
Tabla 4-44- Diálogo Gestionar Actores (Sin actores guardados).....	99
Tabla 4-45 - Diálogo Gestionar Actores (Con actores guardados)	100
Tabla 4-46 - Diálogo Añadir actor (favorito o lista negra).....	100
Tabla 4-47 - Diálogo Actor favorito añadido	100
Tabla 4-48 - Diálogo Actor añadido a lista negra	100
Tabla 4-49 - Diálogo Borrar Actores favoritos	100
Tabla 4-50 - Diálogo Lista negra de actores borrada	100
Tabla 4-51 - Diálogo Gestionar Cines (sin cines guardados)	101
Tabla 4-52 - Diálogo Gestionar Cines (Con cines guardados).....	102
Tabla 4-53 - Diálogo Escoger cine de lista paginada.....	102
Tabla 4-54 - Diálogo Cine favorito añadido	102
Tabla 4-55 - Diálogo Borrar cines favoritos	102
Tabla 4-56 - Diálogo Error se necesita provincia.....	102
Tabla 4-57 - Diálogo Error se necesita cines favoritos.....	104
Tabla 4-58 - Diálogo Mostrar recomendación de sesión paginadas (Sin sesiones)	105
Tabla 4-59 - Diálogo Mostrar recomendación de sesión paginadas	105
Tabla 4-60 - Diálogo Listado paginado de películas.....	107
Tabla 4-61 - Diálogo Ver Película para usuario no identificado	108
Tabla 4-62 - Diálogo Ver Película para usuario identificado sin preferencias	108
Tabla 4-63 - Diálogo Ver Película para usuario identificado con preferencias.....	108
Tabla 4-64 - Diálogo Ver película detallada.....	109
Tabla 4-65 - Diálogo Ver críticas de usuario.....	110

CAPÍTULO 1. Introducción y Objetivos

1.1 Antecedentes

El auge de las tecnologías de la información en el mundo moderno genera la necesidad cada vez mayor de acceder a los datos desde cualquier sitio y a la máxima velocidad posible.

En la actualidad la mayoría de las personas dispone de un dispositivo móvil con conexión a internet (Smartphone, Tablet, phablet...) y cómo podemos observar cada vez existen más aplicaciones, páginas web y empresas que emplean los sistemas de voz para poder controlar sus datos.

Para facilitar su uso, las interfaces diseñadas tienden a la combinación de entradas a través del tradicional teclado y ratón con tecnologías interactivas de entradas por voz. A este tipo de interacción, en la que el usuario tiene más de una forma de acceso a los datos utilizando modos humanos (la voz o habla, gestos, movimientos, ojos...), se le llama multimodal.

La entrada paralela, tanto por voz como por teclado, dota de velocidad al entorno al permitir accesos más rápidos, además de añadir valor a la experiencia general de la interacción. Esta dualidad reduce el esfuerzo excesivo que puede resultar de la interacción con sistemas de modalidad única.

A través de la tecnología VoiceXML se puede conseguir este tipo de iteraciones, ya que VoiceXML es un sistema automático capaz de emular a un ser humano en un diálogo con otra persona, con el objetivo de que el sistema cumpla con una cierta tarea:

- Suministrar información.
- Reserva de viajes, hoteles, restaurantes, etc.
- Automatizar operaciones (entornos domóticos, etc.).
- Banca electrónica.
- Control de máquinas.
- Personas con discapacidades.

VoiceXML es un estándar basado en XML desarrollado por el W3C que permite acceder mediante el habla a aplicaciones Web. Las ventajas de las tecnologías web para el desarrollo de aplicaciones controlables mediante la voz son:

- Integración de servicios de voz y de datos.
- Compatibilidad con otros lenguajes.

Las funcionalidades básicas que utiliza VoiceXML son:

- Reconocimiento de voz.
- Reconocimiento de entrada DTMF.
- Funciones de telefonía.
- Control del flujo del diálogo.
- Grabación de diálogos.
- Salida de voz sintetizada.
- Salida de ficheros de audio.

Por lo tanto, los principales objetivos que se persiguen con la introducción de VoiceXML en las aplicaciones WEB son las siguientes:

- 1) Conseguir que la comunicación con el entorno sea lo más natural posible, es decir, conseguir que la interacción o diálogo sea lo más parecido posible al diálogo que podría establecerse entre dos personas.
- 2) Permitir el uso de aplicaciones en entornos en los que no podrían ser utilizadas debido al uso exclusivo del tradicional sistema de teclado y ratón.
- 3) Facilitar el acceso a las aplicaciones a personas con discapacidades visuales o motoras, favoreciendo la eliminación de barreras arquitectónicas de Internet y el desarrollo para la integración de personas con discapacidades.

1.2 Objetivos

El objetivo de este proyecto final de carrera está dividido en dos partes.

Estas dos partes son el desarrollo de una herramienta que ayude a cualquier desarrollador a desarrollar una aplicación VXML sin tener demasiados conocimientos VXML y una prueba de concepto para comprobar que la herramienta es usable, fácil de usar, la cual será una aplicación que sirva para obtener información sobre películas.

El objetivo fundamental de la herramienta desarrollada es que el usuario por medio de los Controllers y las ViewDatas proporcionadas pueda crear una nueva aplicación VXML sin necesidad de escribir una línea de VXML.

Esta herramienta proveerá las siguientes funcionalidades fundamentales:

- **Creación de menús simples:** Estos menús simples serán opciones cerradas en las cuales el usuario deberá de elegir una opción. Este tipo de menú será utilizado para implementar menús tipo “menú principal” o “páginas finales” en el que se ofrece una información sobre algo y el usuario tiene un número limitado de opciones.

- **Creación de listas paginadas:** Este tipo de página será utilizado para implementar páginas en las que al usuario se le muestra una lista paginada, es decir, una serie de opciones que no caben en una página por lo que se dará en varias páginas. El usuario será informado de las páginas que existen y podrá navegar por ellas.
- **Creación de formularios abiertos:** Estos formularios son páginas en que la opción elegida por el usuario puede ser cualquiera, ya que no se le dan opciones. Estos formularios podrán cargar posibles opciones mediante CSVs.
- **Navegación por sesión:** La infraestructura digital permitirá navegar con sesión para poder controlar si el usuario está identificado o no.
- **Configuración de posibles acciones y parámetros necesarios:** Se configurará que acciones son válidas, así como cuales son por defecto, cuáles de ellos son parámetros obligatorios de cada acción y cuales son parámetros por defecto. Esta configuración será utilizada para la navegación y para la creación de links
- **Sistema de almacenamiento de bases de datos:** La herramienta proveerá de un sistema de Storage con las funciones imprescindibles para manejar una base de datos.

El objetivo de la aplicación además de ser una prueba de concepto de la validez de la infraestructura digital, tendrá como objetivo dar información de películas y de sus proyecciones, la aplicación estará dividida en los siguientes módulos:

- **Módulo de búsquedas:** Este módulo se ocupará de buscar películas, ya sea por título, por actor o por nombre de director.
- **Módulo de cartelera:** Este módulo tendrá la responsabilidad de presentar al usuario las películas que se están emitiendo actualmente en la cartelera. Estas presentaciones pueden mostrarse ordenadas con distintas ordenaciones.
- **Módulo próximos estrenos:** Este módulo tendrá la responsabilidad de mostrar al usuario las películas que se estrenarán en el futuro. Al igual que el módulo anterior, estas presentaciones pueden mostrarse ordenadas con distintas ordenaciones.
- **Módulo Identificación/Registro:** Este módulo tendrá la responsabilidad de poder registrar usuarios, dejar que se autentifiquen y recordarlos una vez identificados.
- **Módulo de preferencias:** Este módulo tiene la responsabilidad de gestionar las preferencias del usuario. Estas preferencias serán:

- Directores favoritos
 - Actores favoritos
 - Géneros de película favoritas
 - Cines favoritos
 - Horarios favoritos para ir al cine
 - Directores que no le gusten
 - Actores que no le gusten
 - Géneros de películas que no le gusten.
- **Módulo de recomendación de películas:** Este módulo tendrá la responsabilidad de buscar las películas que se proyectan en los cines favoritos de un usuario, durante su horario favorito y ordenarlas dependiendo de los gustos del usuario.
- **Módulo de información de la película:** Este módulo tendrá la responsabilidad de ser capaz de mostrar una película al usuario de manera resumida o detallada y ser capaz de mostrar las críticas recibidas por los usuarios.
- **Módulo de obtención de información de FilmAffinity:** Este módulo tiene la responsabilidad de obtener la información recibida por FilmAffinity, para seguidamente poder modelarla al dominio local y utilizarla.

Con estos dos objetivos definidos se quiere conseguir comprobar, si una infraestructura digital que ayude a implementar VXML es viable o no.

Para poder llevar a cabo la implementación de la aplicación y la resolución del proyecto en general, se definieron una serie de objetivos parciales que pueden quedar resumidos en los siguientes puntos:

- Llevar a cabo un estudio de las tecnologías VoiceXML y XHTML+Voice, referidas al desarrollo de interfaces orales unimodales y multimodales, con el fin de mostrar las posibilidades que ofrece, explicar su funcionamiento y poder decidir qué dirección y qué finalidad darle al proyecto.
- Crear una infraestructura digital de navegación VXML válido. Esta infraestructura digital sería básica.
- Crear una aplicación que de información sobre películas. Esta aplicación se haría sin usuarios registrados y solo buscaría películas, mostraría cartelera y mostraría próximos estrenos.
- Añadir registro de usuarios a la aplicación y preferencias.
- Realizar un sistema de recomendación de películas.

1.3 Fases de desarrollo

Las tareas y fases del presente proyecto se agrupan en tres grandes grupos: fase de planificación, fase de desarrollo y fase de documentación.

- En la **fase de planificación** se definen los objetivos y se planifica el curso de acción requerido para lograr los objetivos y el alcance del proyecto.

Para realizar esta fase se ha procedido a realizar un estudio de los sistemas de diálogo (VoiceXML), analizando y recogiendo todos los requisitos necesarios para la realización de la aplicación.

Además, se ha realizado un estudio para determinar las tecnologías necesarias para la correcta implementación de la aplicación.

- En la **fase de ejecución** se realiza la integración de las personas y de otros recursos necesarios para llevar a cabo el plan de gestión del proyecto.

Además, en esta fase se ha incluido la supervisión del avance del proyecto, a fin de identificar las variaciones que pudieran existir respecto del plan de gestión del proyecto definido.

Las tareas que se han realizado durante esta fase han sido las siguientes:

- Análisis y diseño inicial de la aplicación
 - Implementación completa del sistema
 - Pruebas unitarias
 - Pruebas de integración y de sistema
 - Evaluación de la aplicación.
- En la **fase de documentación** se realizan las siguientes tareas:
 - Memoria del Proyecto Final de Carrera
 - Presentación con diapositivas necesaria para la defensa del proyecto
 - Preparación para la defensa final del Proyecto Final de Carrera.

1.4 Planificación temporal

En este punto se pasan a detallar las tareas contenidas en las fases descritas en el apartado anterior, así como la duración de cada una de ellas.

Además, se presenta el diagrama de Gantt con la planificación temporal de todas las tareas y la fecha de inicio del proyecto.

- **Fase 1: Planificación.**
 - Estudio de los sistemas de diálogo (VoiceXML).
 - Duración: 15 días.
 - Planificación y análisis de requisitos de la aplicación.
 - Duración: 10 días.
 - Estudio de las tecnologías necesarias.
 - Duración: 5 días.
- **Fase 2: Ejecución.**
 - Análisis y diseño inicial.
 - Duración: 15 días.
 - Implementación completa del sistema.
 - Duración: 60 días.
 - Pruebas unitarias.
 - Duración: 10 días.
 - Pruebas de integración y de sistema.
 - Duración: 10 días.
 - Evaluación de la aplicación.
 - Duración: 15 días.
- **Fase 3: Documentación.**
 - Memoria del Proyecto Final de Carrera.
 - Duración: 30 días.
 - Presentación con diapositivas.
 - Duración: 10 días
 - Preparación de la defensa.
 - Duración: 10 días.

1.4.1 Desglose de tareas y Diagrama de Gantt

En este punto se muestra en detalle el desglose de las tareas indicadas en el punto anterior (Figura 1.1) y el diagrama de Gantt (Figura 1.2), que posibilita una visión global de la planificación del proyecto y de las distintas etapas que lo componen, así como su duración global.

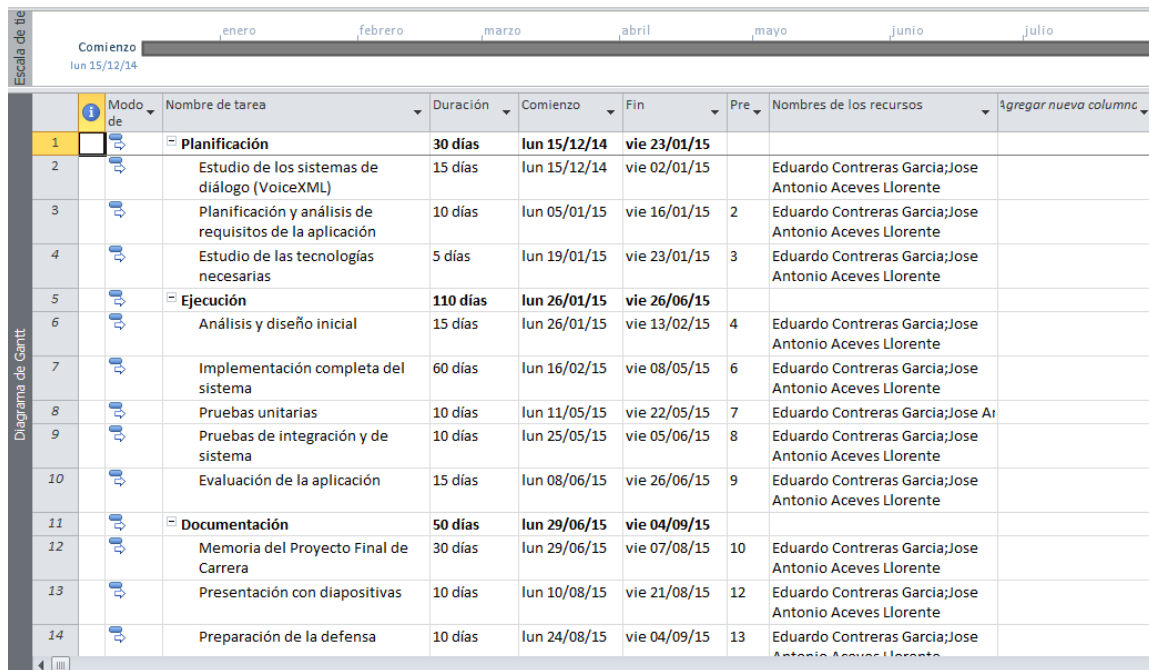


Figura 0-1 - Desglose de las tareas y planificación temporal del proyecto

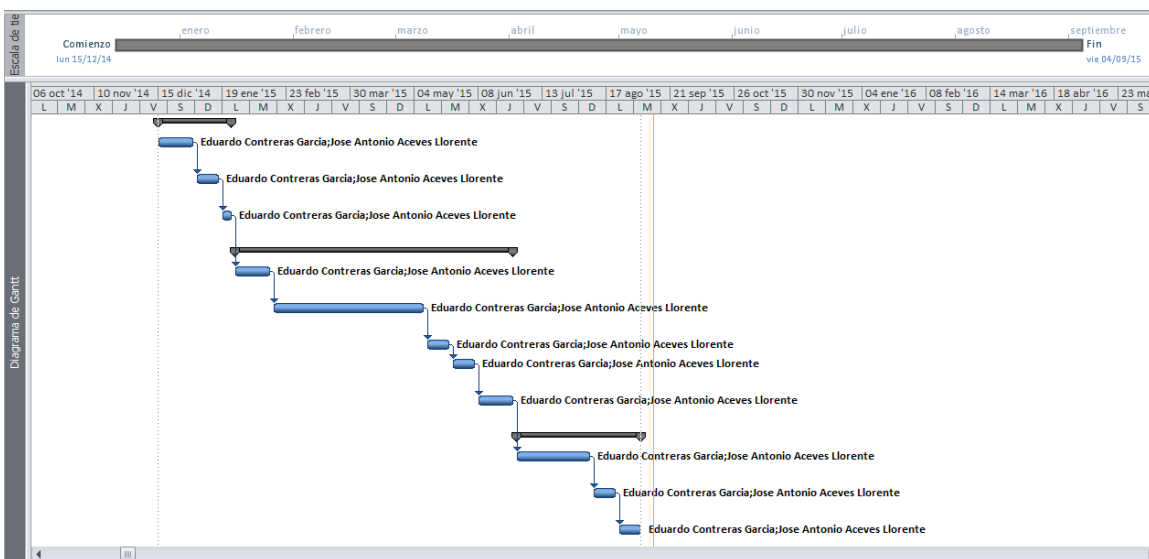


Figura 0-2 - Diagrama de Gantt con la planificación temporal del proyecto

1.5 Medios y documentación empleados

Los medios empleados para la realización del Proyecto Final de Carrera han sido los siguientes:

- **Dispositivos Hardware:**
 - 2 ordenadores portátiles.
 - Micrófono portátil externo.
 - 2 auriculares con micrófono.
 - Periféricos habituales (teclado, ratón, pantalla).
 - Disco duro externo para almacenaje y copias de seguridad
 - Servidor web (<http://maximus.freewha.com/>)
- **Aplicaciones Software:**
 - Plataforma Voxeo.
 - Skype.
 - Paquete Microsoft Office 2010.
 - Microsoft Project 2010.
 - Editor de programación PHPStorm.
 - Editor de programación Notepad ++.
 - Panel de control del servidor web (<http://maximus.freewha.com/>)
 - PhpMyAdmin (Para la creación y gestión de la Base de Datos).
 - Adobe Acrobat Reader (Lector PDF)
 - Navegador Chrome

Para la documentación utilizada durante la realización del proyecto, se han empleado numerosos libros, tutoriales y artículos del ámbito de las interfaces de diálogo y sobre los lenguajes de computación oral (VoiceXML). Además se han utilizado diversas fuentes de internet (tutoriales, manuales, foros) y diversos manuales de los distintos lenguajes de programación utilizados (PHP). Toda esta documentación se encuentra detallada en el apartado de Bibliografía.

1.6 Presupuesto

Teniendo en cuenta la planificación temporal de las tareas (punto anterior) y los recursos utilizados para la elaboración del proyecto, en este punto se va a realizar la elaboración de un presupuesto donde se indicará los gastos realizados en materiales y herramientas, así como el coste de las horas dedicadas a la realización de la aplicación de los recursos humanos.

1.6.1 Recursos

Para la realización del presente proyecto se han utilizado los siguientes recursos:

- **Recursos Hardware:**

- 2 ordenadores portátiles: **2000€**
- Micrófono portátil externo: **0€**
- 2 auriculares con micrófono: **100€**
- Periféricos habituales (teclado, ratón, pantalla): **0€**
- Disco duro externo para almacenaje y copias de seguridad: **50€**
- Servidor web (<http://maximus.freewha.com/>): **0€**

- **Recursos Software:**

- Plataforma Voxeo: **0€**
- Skype: **0€**
- Paquete Microsoft Office 2010/ Microsoft Project 2010: **250€**
- Editor de programación PHPStorm: **0€**
- Editor de programación Notepad ++: **0€**
- Panel de control del servidor web (<http://maximus.freewha.com/>): **0€**
- PhpMyAdmin (Para la creación y gestión de la Base de Datos): **0€**
- Adobe Acrobat Reader (Lector PDF): **0€**
- Navegador Chrome: **0€**

- **Recursos humanos:**

Para la realización de este proyecto han participado tres personas:

- Director del proyecto.
- Dos desarrolladores.

Las horas dedicadas al proyecto han sido de una media de 8 horas diarias, contemplando semanas de 7 días.

Coste de un Ingeniero: 30 €/hora.

1.6.2 Resumen de costes

Coste Recursos Humanos

El coste de recursos humanos de la aplicación se detalla en la siguiente tabla:

TAREA	DEDICACIÓN (Días)	IMPORTE
Fase 1: Planificación	30	14400€
Fase 2: Ejecución	110	52800€
Fase 3: Documentación.	50	24000€
TOTAL:	190	91200€

Tabla 0-1 - Detalle de Costes de Recursos Humanos del Proyecto

Coste Total del Proyecto

El coste total del proyecto se detalla en la siguiente tabla:

CONCEPTO	IMPORTE
Recursos Humanos	91200€
Recursos Hardware	2150€
Recursos Software	250€
Subtotal:	93600€
(21% IVA)	19656€
TOTAL	113256€

Tabla 0-2 - Detalle del Coste Total del Proyecto

El presupuesto total del proyecto asciende a la cantidad de **CIENTO TRECE MIL DOSCIENTOS CINCUENTA Y SEIS EUROS**.

Madrid a 1 de Julio de 2011

Fdo. José Antonio Aceves Llorente

1.7 Estructura de la memoria

A continuación se incluye un breve resumen del contenido de cada capítulo de la memoria, con el objetivo de facilitar la lectura de la memoria:

- **Capítulo 1: Introducción y objetivos.** Capítulo donde se establece el propósito y los objetivos de todo el contenido del Proyecto Final de Carrera. Además, incluyen las fases de desarrollo, la planificación temporal, el presupuesto, los medios y documentación empleados y la estructura de la memoria.

El apartado de presupuesto, contiene un análisis de los costes del diseño y desarrollo del proyecto, detallando el coste de personal y del material necesario para llevar a cabo su realización.

- **Capítulo 2: Estado del Arte.** Capítulo donde se realiza un estudio completo de los sistemas de diálogo, se analiza en detalle el lenguaje estándar VoiceXML y se presenta la plataforma Voxeo utilizada para la implementación de la aplicación.
- **Capítulo 3: Descripción del sistema.** Capítulo donde se realiza una visión global del sistema y de la infraestructura digital para desarrollar aplicaciones VoiceXML.
- **Capítulo 4: Descripción detallada del sistema.** En este capítulo se describe detalladamente cada uno de los módulos de los que consta el sistema desarrollado. Para cada uno de los módulos se explica sus funcionalidades, arquitectura y escenarios de uso.
- **Capítulo 5: Evaluación de la aplicación.** En este capítulo se razona la importancia de la evaluación de los sistemas de diálogo y a través de las valoraciones de los usuarios recogidas mediante un cuestionario se realiza la evaluación del sistema desarrollado.
- **Capítulo 6: Conclusiones y trabajo futuro.** Se exponen las principales ideas, cuestiones y conclusiones derivadas de la realización del proyecto, así como las posibles líneas de investigación que puedan surgir a partir de la realización de este proyecto.
- **Glosario.** Apartado donde se recopilan los principales términos y conceptos técnicos utilizados en la memoria con el objetivo de facilitar su comprensión al lector.
- **Bibliografía.** Apartado donde se reflejan las citas bibliográficas que se han consultado para la realización tanto del proyecto como de la memoria.

CAPÍTULO 2. Estado del Arte

En este apartado daremos a conocer una visión global del estado del arte sobre la tecnología VXML, la cual ha sido la elegida para realizar este proyecto.

2.1 Introducción a los sistemas de diálogo

Los sistemas de diálogo basados en procesamiento del habla son sistemas informáticos que reciben como entrada frases del lenguaje natural expresadas de forma oral y generan como salida frases del lenguaje natural expresadas asimismo de forma oral [\[1\]](#)

La finalidad de estos sistemas es emular el comportamiento inteligente de un ser humano en un diálogo con otra persona, con el objetivo de que se realice una cierta tarea [\[2\]](#).

Un sistema de diálogo ideal reconocería el habla espontánea, comprendería enunciados sin restricciones de contenido, proporcionaría respuestas con sentido, gramaticalmente bien formadas y pragmáticamente adecuadas, respondería con voz completamente natural y sería multimodal [\[3\]](#).

2.2 Evolución de los sistemas de diálogo

La tecnología a lo largo de los años ha evolucionado de forma significativa en el objetivo de lograr que el esquema de comunicación de los seres humanos consiga una representación efectiva a través de la conjunción de una aplicación informática y su desarrollo en una interfaz sencilla de manejar, para que el usuario final pueda llevar a cabo las tareas cotidianas desde su casa, simplemente utilizando la voz como plataforma para la consecución de su objetivo: la compra de una entrada o billete de avión, la redacción de una carta, o bien, la consulta de los datos bancarios [\[4\]](#).

2.2.1 Primeros sistemas de diálogo (años 60)

Durante los años 60 se desarrollaron un conjunto de sistemas con la intención de proporcionar interfaces de lenguaje entre los ordenadores y los humanos. La mayor parte de estos desarrollos usaba diferentes técnicas para procesar el lenguaje y producir resultados aceptables.

BASEBALL

BASEBALL (Green et al., 1963), era un sistema de “pregunta-respuesta” capaz de contestar a preguntas sobre fechas, lugares, equipos y puntuaciones de partidos de béisbol.

Para ilustrar la naturaleza ad hoc de parte del procesamiento del lenguaje, el significado de la palabra “quién” en el diccionario del sistema, era dado como “Equipo = ?”, un significado específico del dominio que no podría aplicarse a otros dominios [5].

STUDENT

STUDENT, era un programa capaz de solucionar problemas de álgebra planteados con un lenguaje natural, como por ejemplo:

1. “María tiene el doble de años que tenía Ana cuando María tenía los mismos años que Ana tiene ahora.
2. Si María tiene ahora 24 años, ¿cuántos años tiene Ana?”

El programa convertía el lenguaje natural en un conjunto de ecuaciones, desglosándolo en patrones simples y buscando palabras y frases que podían ser sustituidos por variables y expresiones aritméticas.

Este sistema demostró algunas de las características más interesantes que vaticinaron desarrollos posteriores, como por ejemplo, que el sistema fuera capaz de tratar toda la conversación en conjunto, y no frases aisladas. Otra característica era que el sistema usaba un almacenamiento de conocimiento general, obteniendo más información del usuario.

En cualquier caso, sistemas como BASEBALL o STUDENT, se basaban generalmente en técnicas que podían ser aplicadas con éxito dentro del dominio en cuestión, pero sin embargo no podían generalizarse a otros dominios. Además, dichas técnicas no estaban basadas en aproximaciones al diálogo más teóricamente motivadas [6].

2.2.2 Procesamiento del lenguaje natural (Años 70)

Durante los años 70 y principios de los 80, comenzaron a aparecer dichas aproximaciones, basadas en la investigación sobre el procesamiento del lenguaje natural y la inteligencia artificial.

El procesamiento de lenguajes naturales —abreviado PLN, o NLP del idioma inglés Natural Language Processing— es un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano. El PLN se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas y máquinas por medio de lenguajes naturales. El PLN no trata de la comunicación por medio de lenguajes naturales de una forma abstracta, sino de diseñar mecanismos para comunicarse que sean eficaces computacionalmente —que se puedan realizar por medio de programas que ejecuten o simulen la comunicación—. Los modelos aplicados se enfocan no sólo a la comprensión del lenguaje de por sí, sino a aspectos generales cognitivos humanos y a la organización de la memoria. El lenguaje natural sirve sólo de medio para estudiar estos fenómenos. Hasta la década de 1980, la mayoría de los sistemas de PNL se basaban en un

complejo conjunto de reglas diseñadas a mano. A partir de finales de 1980, sin embargo, hubo una revolución en PNL con la introducción de algoritmos de aprendizaje automático para el procesamiento del lenguaje.

SHRDLU

SHRDLU Es un programa de computadora para la comprensión del lenguaje natural. Fue desarrollado por Terry Winograd en el MIT entre los años 1968 y 1970. Fue implementado con los lenguajes de programación Micro Planner y LISP para una máquina DEC PDP-6 con terminal gráfico DEC. Posteriormente en la Universidad de Utah le fue añadido un mundo en 3D donde se representaba gráficamente los objetos conocidos por el programa.

SHRDLU permite al usuario interactuar mediante frases simples de inglés. El usuario debe instruir a SHRDLU como mover los objetos de su entorno. Este entorno fue llamado small blocks word (pequeño mundo de bloques). Contenía objetos tales como bloques, conos, bolas y otras formas geométricas. Lo que hizo único a SHRDLU, es que con la combinación de los conceptos anteriores se podía realizar una simulación de la comprensión computacional del lenguaje.

El entorno o mundo de bloques era lo suficientemente simple para poder ser descrito completamente (objetos, características y posiciones) usando un conjunto reducido de palabras en inglés (50 aproximadamente). Nombres como bloque, cono, y pirámide, verbos como pon, mueve y deja y adjetivos como pequeño, grande y rojo. Las combinaciones posibles de este lenguaje básico permitían al programa realizar lo que el usuario le ordenaba.

SHRDLU incluía una memoria básica de trabajo que contenía el contexto de la conversación. Por ejemplo, se le podía decir: Pon el cono verde encima del bloque rojo, y a continuación: Coge el cono. El cono que queremos coger debe significar el que nos hemos referido anteriormente y no cualquier otro que este en el entorno. SHRDLU debía buscar el último contexto más próximo en el cual hemos hablado de El cono. Esto permitía en muchas ocasiones poner o no adjetivos adicionales. Por otra parte se podía interrogar acerca del historial de acciones, por ejemplo: Dime que has cogido antes del cono [7].

Consultas a Bases de Datos de Lenguaje Natural

Los sistemas que podían procesar consultas de lenguaje natural y traducirlas a un lenguaje formal de consulta a base de datos, constituyeron uno de los primeros éxitos del procesamiento del lenguaje natural. El sistema LUNAR (Woods et al., 1972), que podía contestar preguntas formuladas en lenguaje natural accediendo a una base de datos de rocas lunares, fue uno de estos primeros interfaces aparecidos a finales de los años 60.

ELIZA

ELIZA es un programa informático diseñado en el MIT entre 1964 y 1966 por Joseph Weizenbaum. Eliza fue uno de los primeros programas en procesar lenguaje natural. El mismo parodiaba al psicólogo Carl Rogers e intentaba mantener una conversación de texto coherente con el usuario.

ELIZA funciona buscando palabras clave en la frase escrita por el usuario y responde con una frase modelo registrada en su base de datos.

A veces, las respuestas de este sistema conversacional resultaban tan convincentes que al conversar por primera vez con él, algunas personas creyeron que en verdad estaban dialogando con un interlocutor humano. Sin embargo, esto tuvo un límite cuando el usuario encontraba mensajes implícitos o subliminales en las respuestas del sistema que en verdad no existían o no intentaban significar eso. Llegado este punto, la conversación se volvía incoherente.

Cuando se escribía determinada frase que el sistema no conocía, el mismo reformulaba dicha expresión a modo de pregunta o reflexión [\[8\]](#).

PARRY

Este sistema, contemporáneo a ELIZA, simulaba a un paciente paranoico, y en ciertas pruebas los psiquiatras fueron incapaces de distinguir entre locuciones realizadas por PARRY y paranoicos reales.

Disponía de un conjunto de más de 6000 patrones contra las que se contrastaba cualquier entrada. Mientras que ELIZA no tomaba la iniciativa en la conversación, sino que reflejaba lo que había dicho el usuario, PARRY era capaz de mantener una conversación ya que siempre tenía algo que decir [\[9\]](#).

CONVERSE

Mediante un personaje (Catherine), y a través de una serie de scripts que cubrían un amplio abanico de 80 temas (incluyendo algunos tan variados como crimen, racismo, religión, teléfonos móviles, la teleserie “Los Simpson”, o el aborto), es capaz de mantener el control de una conversación realizando y respondiendo preguntas [\[20\]](#). Mediante un sistema de cálculo de “pesos” de palabras, podía determinar si el sistema debía preguntar o responder.

2.2.3 Tecnologías del Habla

Los sistemas de lenguaje natural previamente descritos, están basados en entradas y salidas de información.

Esto se debía en parte al hecho de que la tecnología del habla no se había desarrollado lo suficiente como para manejar el complejo lenguaje natural usado en estos sistemas. También fue debido a que prácticamente no hubo interacción entre el mundo de la inteligencia artificial y el mundo de las tecnologías del habla hasta mediados de los años 80.

La investigación sobre el reconocimiento del habla en los años 60 se centró en sistemas caracterizados por:

1. Reconocimiento dependiente del locutor: el sistema tenía que estar entrenado para reconocer el habla de un usuario individual
2. Reconocimiento discreto de palabras: el hablante tenía que hacer una pausa entre cada palabra para que el sistema pudiera identificar los límites de dichas palabras.
3. Vocabularios pequeños de menos de 50 palabras.

Con el objetivo de superar estas limitaciones, de 1971 a 1976 se creó un importante programa de investigación, llamado Investigación para la Comprensión del Habla (Speech Understanding Research, SUR), y patrocinado por la Agencia de Investigación de Proyectos Avanzados (Advanced Research Projects Agency, ARPA, 1994). Se pretendía que los sistemas desarrollados fueran capaces de reconocer las locuciones de varios hablantes, usando un vocabulario de 1000 o más palabras. Uno de ellos, HARPY, de la Universidad Carnegie Mellon (capaz de reconocer más de 1000 palabras con un ratio de error del 5%), fue uno de los primeros en usar tipos de modelado basados en estadísticas, usados actualmente en gran número de sistemas de reconocimiento del habla comerciales y de investigación.

Los trabajos posteriores se basan en el desarrollo de modelos estadísticos robustos, capaces de manejar locuciones continuas con amplio vocabulario, y superando dificultades como habla a través de teléfono, conversaciones con ruido de fondo. Como resultado, la investigación reciente en la tecnología del diálogo hablado ha reunido conceptos del Procesamiento del Lenguaje Natural con los de la Inteligencia Artificial, que durante las décadas previas se habían desarrollado de forma independiente.

La investigación en la tecnología del diálogo hablado surgió a finales de los años 80, como resultado de los proyectos: el programa de Sistema de Lenguaje Hablado DARPA, en los EE.UU. y el programa Esprit SUNDIAL en Europa. El programa DARPA estaba relacionado con el dominio de los Servicios de Información de Vuelos (Air Travel Information System, ATIS), y su objetivo era investigar y desarrollar tecnologías relacionadas con el reconocimiento del habla, usando el dominio de la reserva de vuelos a través de un canal telefónico. En la actualidad, el corpus de diálogos de ATIS, sigue siendo usado por desarrolladores y examinadores de sistemas de diálogos.

Seneca

Se trata de un sistema orientado al uso de dispositivos móviles mediante el cual, una serie de aplicaciones de diferente ámbito (navegación, entretenimiento, comunicación, etc.) permiten diálogos entre el usuario y la máquina. [\[10\]](#).

Utiliza reducción de ruido, reconocedor del habla, así como técnicas de procesamiento del diálogo [\[10\]](#).

Chat

Este Sistema conversacional de ayuda a las tareas automotrices está destinado al control de dispositivos dentro de un automóvil [\[11\]](#).

Proporciona un interfaz basado en la voz para tareas tan variadas como escuchar música, encontrar un restaurante, o navegar por internet. Todo ello a la vez que el usuario está conduciendo.

Discoh

Desarrollado gracias a fondos públicos, implementa un sistema de diálogo hablado destinado a servicios de conferencia [\[12\]](#). Mediante este sistema, se recopila un corpus de diálogos hombre-máquina de iniciativa mixta.

Los usuarios del sistema pueden llamar a un número de teléfono y obtener información sobre la conferencia, como presentación de impresos, programa, lugar, alojamientos, costes, etc.

El corpus recopilado se usa para entrenamiento, evaluación y comparación de modelos estadísticos; es orientado al lenguaje natural y a tareas, extensible y generalizable, recopilado mediante una tecnología comercial y de investigación de estado del arte, y finalmente está disponible abiertamente para los investigadores

Conquest

Al igual que Discoh, está orientado a conferencias, y se trata de un sistema de código abierto que proporciona información técnica del programa durante las mismas [\[13\]](#). Usa una infraestructura transparente, modular y abierta, y su objetivo es el permitir la investigación aplicada en interfaces de lenguaje hablado.

El dominio de la conferencia, es una buena plataforma para la investigación aplicada, ya que permite la reorganización y evaluación periódica con una verdadera base de usuarios reales.

Radiobot-CFF

De ámbito militar, es un sistema de diálogo dedicado al entrenamiento de soldados en la simulación de Llamada a Fuego (Call for fire, CFF), para la correcta solicitud de fuego de artillería en las misiones [\[14\]](#).

2.2.4 Retos actuales

Aunque es complicado realizar predicciones sobre la evolución de los sistemas de diálogo en los próximos años, se pueden establecer una serie de retos a cumplir por los mismos:

1. Desarrollo de nuevas metodologías que permitan adaptar el funcionamiento de los sistemas de diálogo, teniendo en cuenta las necesidades y preferencias de cada usuario (sistemas adaptativos).
2. La comunicación persona a persona es multimodal, combinando diferentes canales, por lo tanto, el desarrollo de sistemas multimodales y su evaluación en diferentes entornos como redes sociales, mundos virtuales, etc., será de gran importancia.
3. Diseño de sistemas multitarea.
4. Análogamente al reconocimiento del habla, será de gran utilidad el reconocimiento de emociones. Es fundamental el diseño de sistemas que tienen en cuenta el comportamiento emocional del usuario y que son igualmente capaces de generar respuestas que sinteticen estas emociones.

2.3 Arquitectura de los sistemas de diálogo

El esquema utilizado para el desarrollo de sistemas de diálogo suele englobar una serie de módulos genéricos que deben coordinarse para responder a los requerimientos del usuario [3] [15].

En la figura 2-1 se muestra la arquitectura básica de un sistema de diálogo:

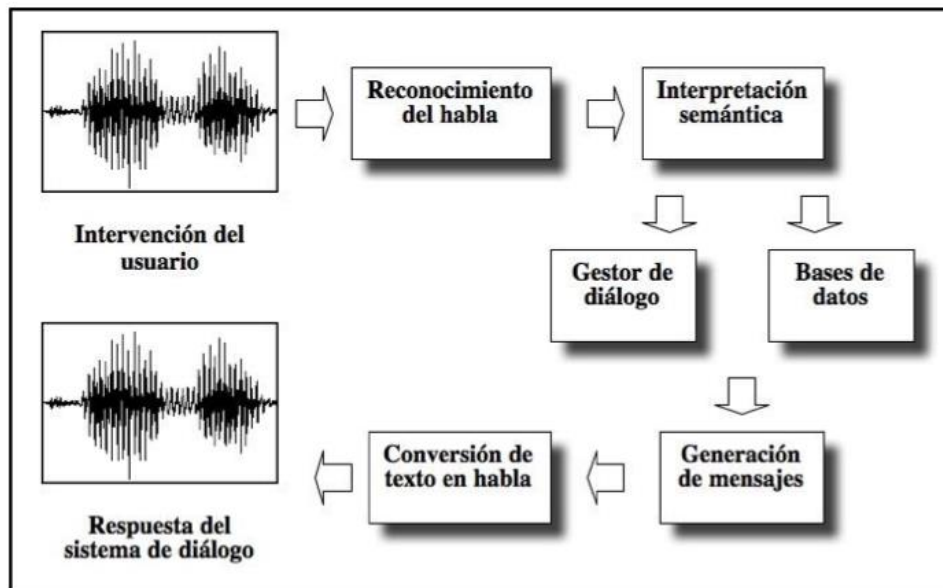


Figura 0-1 - Arquitectura de un sistema de diálogo

La descripción de los módulos y sus principales características vistas en la figura anterior se explican a continuación:

Módulo de Reconocimiento Automático del Habla (RAH): Procesa la voz de usuario y la transforma en la secuencia de palabras reconocidas con mayor probabilidad.

Módulo de Interpretación semántica (PLN): Extrae el significado de las palabras reconocidas en el módulo anterior, obteniendo la representación semántica (significado) de la frase y expresándolo en términos de un lenguaje especificado para la tarea.

Gestor del diálogo (GD): Decide qué paso debe dar el sistema tras cada intervención del usuario. Puede decirse que este es el módulo fundamental del sistema, pues su finalidad es lograr que la interacción con el usuario sea lo más cómoda e “inteligente” posible. Para ello, se basa en la interpretación semántica de la petición del usuario, la historia del proceso de diálogo, la información disponible en ese punto, el estado actual del sistema, la información obtenida de la base de datos, la **estrategia definida, etc.**

Módulo de Generación del Lenguaje Natural (GLN): Tiene como función la generación de una frase, gramaticalmente correcta y en un lenguaje lo más cercano posible al lenguaje natural, que transmita el mensaje generado **por el gestor de diálogo**.

Módulo de Conversión Texto Habla (CTH): Transforma la frase de respuesta en señal de audio.

2.4 Clasificación de los sistemas de diálogo

Los sistemas de diálogo se pueden clasificar según distintos criterios [\[2\]](#):

2.4.1 Iniciativa del diálogo

Sistemas de diálogo guiados por el sistema: La interacción se realiza mediante turnos concretos y fijados entre pregunta y respuesta. Se restringen las iniciativas del usuario.

Sistemas de iniciativa mixta: Aceptan las interrupciones y negociaciones por parte del usuario, reparten de forma equilibrada el turno de palabra e incorporan mecanismos de detección de incoherencias gramaticales.

Sistemas de diálogo guiados por el usuario: Es el usuario el que lleva la iniciativa de diálogo en cualquier instante de la interacción, permitiéndose al usuario que utilice lenguaje natural y no delimitando los mensajes del sistema (por ejemplo, ¿en qué puedo ayudarle?).

2.4.2 Dependencia del hablante

Dependiente del hablante: Un sistema dependiente del hablante está desarrollado para funcionar con un solo hablante. Estos sistemas, normalmente, son más fáciles de implementar, más baratos y más precisos, pero no tan flexibles como los sistemas adaptables al hablante o los sistemas independientes del hablante.

Independiente del hablante: Un sistema independiente del hablante está desarrollado para funcionar para cualquier hablante de un determinado tipo (por ejemplo, inglés americano). Estos sistemas son los más complicados de desarrollar, los más caros y la precisión es menor que la de los sistemas dependientes del hablante. Sin embargo, son más flexibles.

2.4.3 Tipo de comunicación

- **Unimodal (oral):** Solo se utiliza el habla.
- **Multimodal:** Se utilizan varios canales de comunicación. Por ejemplo, dispositivos de entrada como el habla, el teclado, el ratón, el micrófono, la cámara, una pantalla o un PDA y canales de salida para proporcionar información como voz, texto, gráficos o imágenes.

2.4.4 Tipo de idioma

- **Monolingüe:** Permiten interacción solo mediante un idioma.
- **Multilingüe:** Permiten interacción mediante varios idiomas. Por ejemplo, castellano e inglés.

2.4.5 Tipo de discurso

Reconocimiento de palabras aisladas: En el reconocimiento de palabras aisladas el patrón acústico de cada palabra del vocabulario se almacena como una secuencia temporal de características derivadas, usando LPC, análisis de banco de filtros o alguna otra técnica de análisis del lenguaje. El reconocimiento se lleva a cabo comparando el patrón acústico de la palabra a reconocer con los patrones almacenados, seleccionando la palabra que mejor encaje con la palabra desconocida.

Reconocimiento de palabras conectadas: En el reconocimiento de palabras conectadas, la entrada hablada es una secuencia de palabras de un vocabulario específico, y el reconocimiento se lleva a cabo basándose en la coincidencia de palabras de referencia aisladas. Ejemplos de esto son las cadenas de dígitos conectados donde el vocabulario es un conjunto de 10 dígitos, o el reconocimiento de letras conectadas, donde el vocabulario es el conjunto formado por el abecedario. No hay que confundir esto con el reconocimiento continuo de voz, donde el reconocimiento se basa en unidades lingüísticas denominadas fonemas, sílabas, etc., lo que supone separar la voz en estas unidades y etiquetarlas subsecuentemente.

Reconocimiento continuo: Un sistema de reconocimiento continuo funciona sobre un lenguaje en el que las palabras están conectadas, es decir, no están separadas por pausas. El lenguaje continuo es más difícil de tratar debido a la variedad de efectos. Primero, es difícil encontrar el comienzo y el final de las palabras. Otro problema es la coarticulación. La generación de cada fonema se ve afectada por la generación de los fonemas adyacentes, y de modo parecido, el comienzo y final de las palabras se ven afectados por las palabras que les preceden y suceden. El reconocimiento del lenguaje continuo también se ve condicionado por la frecuencia de habla (un discurso rápido suele ser más difícil).

Reconocimiento discreto: Un sistema de reconocimiento discreto funciona con palabras simples, necesitando de una pausa entre la dicción de cada palabra. Esta es la forma más sencilla de reconocimiento para llevar a cabo ya que los puntos de finalización son más fáciles de encontrar y la pronunciación de una palabra no afecta a las demás. De este modo, y ya que las ocurrencias de las palabras son más consistentes, es más fácil reconocerlas.

2.4.6 Adaptación

Sistemas adaptativos: El sistema es capaz de aprender nuevas estrategias comunicativas en función del comportamiento del usuario.

Una línea de investigación actual consiste en la incorporación de las emociones a los sistemas de diálogo. Para ello, el análisis del comportamiento del usuario no debe incluir únicamente las emociones básicas sino también todos sus estados emocionales y cambios en el comportamiento. Por ejemplo, si se detecta que el usuario puede estar enfadado, se inician estrategias de recuperación o se transfiere la llamada a un operador humano, o si se detecta algo similar a la alegría, se aprovecha para ofrecer un nuevo producto.

2.5 VoiceXML

VoiceXML (VXML) es el formato estándar XML de W3C que permite la interacción persona-máquina mediante diálogos por voz. Su desarrollo y despliegue es análogo al desarrollo y despliegue de aplicaciones web (HTML). De esta manera al igual que un documento HTML es interpretado por un navegador web, un fichero VXML es interpretado por un intérprete VXML.

El lenguaje VoiceXML es un lenguaje basado en etiquetas, mediante las cuales se indica al intérprete que debe hacer (leer texto, reconocer palabras, gestionar el diálogo con el usuario o reproducir audio).

El protocolo HTTP es el más usado para la obtención de páginas VoiceXML. Las aplicaciones pueden usar, por un lado, ficheros VoiceXML estáticos, o bien usar un servidor web o de aplicaciones para generar los ficheros VoiceXML de forma dinámica. En el caso que nos ocupa y que corresponde al desarrollo de este Proyecto de Fin de Carrera, la aplicación solo usa ficheros VXML dinámicos.

Actualmente este estándar se encuentra en su versión 2.1, aunque está prevista próximamente una nueva versión 3.0, de la que ya puede consultarse un borrador.

Las principales funcionalidades que ofrece VoiceXML son la salida de voz sintetizada, salida de ficheros de audio, reconocimiento de entradas por voz y DTMF, grabación de diálogos, control del flujo del diálogo y funciones de telefonía.

El funcionamiento básico consiste en capturar la entrada del usuario (mediante voz o DTMF), asignar valores a una serie de variables y tomar decisiones en función de dichas variables.

En la siguiente figura se puede ver un diagrama de interacción del intérprete de VoiceXML con un servidor web:

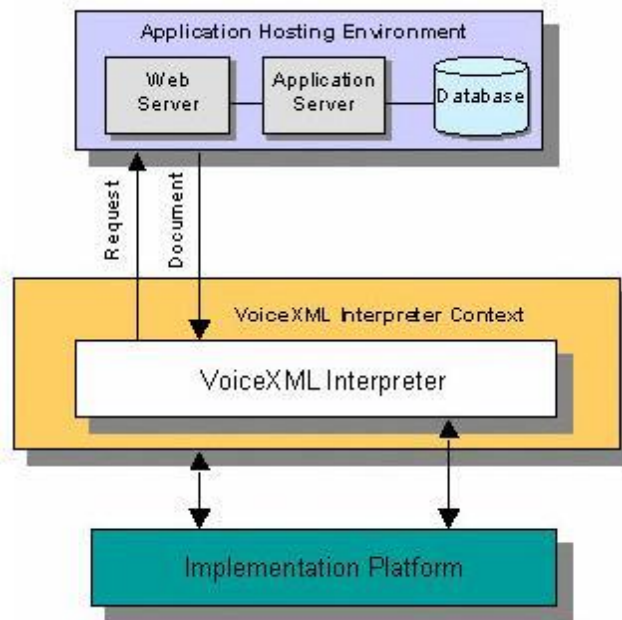


Figura 0-2 - Diagrama de interacción de un intérprete VXML con un servidor web

2.5.1 Conceptos básicos de VoiceXML

Un documento VoiceXML puede considerarse como una máquina de estados finitos. El usuario está siempre en un estado, o diálogo, en un determinado momento. Cada diálogo determina el siguiente diálogo al que hacer la transición. Las transiciones se especifican mediante URIs (Uniform Resource Identifier) [\[15\]](#), que definen el siguiente documento y diálogo a utilizar.

La ejecución se termina cuando un diálogo no especifica un sucesor, o si tiene un elemento que hace salir explícitamente de la conversación.

Los conceptos básicos referentes a VoiceXML son los siguientes:

Diálogos y subdiálogos

Hay dos tipos de diálogos: formularios y menús.

Los formularios definen una interacción que recoge los valores de un conjunto de variables. Cada campo puede especificar una gramática que define las entradas permitidas para ese campo.

Un menú presenta al usuario una amplia gama de opciones y, a continuación, transiciones a otro diálogo basado en la elección elegida.

Un subdiálogo es como una llamada a función, que proporciona un mecanismo para invocar una nueva interacción y volver al formulario original. Las instancias de variables, las gramáticas y la información de estado se guardan y están disponibles al volver al documento inicial.

Cuando los subdiálogos son invocados añaden un nuevo contexto de ejecución. El subdiálogo puede ser un nuevo diálogo dentro del documento existente o un nuevo diálogo dentro de un nuevo documento. Los subdiálogos pueden estar compuestos de varios documentos.

Sesión

Una sesión comienza cuando el usuario comienza a interactuar con el intérprete de contexto VoiceXML, continúa mientras los documentos son cargados y procesados, y termina cuando se solicita por el usuario, por un documento o por el intérprete de contexto VoiceXML.

Aplicación

Una aplicación es un conjunto de documentos que comparten el mismo documento raíz. Este documento raíz se carga cuando el usuario accede a algún diálogo contenido en un documento contenido en él, el documento raíz permanecerá cargado mientras el usuario navegue por los distintos diálogos contenidos en los documentos de la aplicación. Este documento finalmente será descargado una vez el usuario acceda a un diálogo que se encuentre fuera de esta aplicación.

Cuando una aplicación tiene un solo documento, la ejecución del documento comienza por defecto en el primer diálogo. Según se ejecuta cada diálogo, se determina el siguiente diálogo. Cuando el diálogo no especifica un diálogo sucesor, se detiene la ejecución del documento.

Si una aplicación tiene varios documentos, cada documento se ejecuta como una aplicación aislada. En los casos en los que se quiera tener varios documentos para trabajar conjuntamente como una única aplicación, se debe seleccionar un documento como documento raíz de la aplicación y el resto como documentos hoja. Cada documento

hoja nombra al documento raíz en su elemento <Vxml>.

Cada vez que el intérprete tenga que cargar y ejecutar un documento hoja, primero cargará el documento raíz de la aplicación si no está ya cargado. El documento raíz de la aplicación permanecerá cargado hasta que el intérprete tenga que cargar un documento que pertenezca a otra aplicación. Por lo tanto siempre se mantendrá una de las dos condiciones siguientes durante la interpretación:

- a. El documento raíz de la aplicación se carga y el usuario está ejecutando en él. No hay ningún documento hoja.
- b. El documento raíz de la aplicación y un sólo documento hoja son cargados. El usuario está ejecutando en el documento hoja.

Si hay una cadena de subdiálogos definida en documentos separados, puede haber más de un documento hoja cargado aunque la ejecución se realizará sólo en uno de estos documentos.

Cuando un documento hoja hace que se cargue un documento raíz, ninguno de los diálogos en el documento raíz se ejecutan. La ejecución comienza en el documento hoja.

Una de las ventajas de las aplicaciones multidocumento es que las variables y las propiedades del documento raíz están disponibles para su uso por los documentos hoja, de esta forma la información se puede compartir y conservar.

Gramáticas

Cada diálogo puede tener asociadas una o más gramáticas de voz o de DTMF.

En las aplicaciones dirigidas por ordenador, las gramáticas están activas sólo cuando el usuario está en ese diálogo.

En las aplicaciones de iniciativa mixta, donde el usuario y la máquina se alternan en la determinación de qué hacer a continuación, algunos de los diálogos están marcados para hacer sus gramáticas activas incluso cuando el usuario se encuentra en otro diálogo en el mismo documento, o en otro documento cargado en la misma aplicación. En esta situación, si el usuario dice algo que coincida con las gramáticas activas del otro diálogo, se ejecuta la transición a ese otro diálogo. La iniciativa mixta agrega flexibilidad y potencia para las aplicaciones de voz.

El objetivo de estas gramáticas es reconocer las respuestas dichas por el usuario de la aplicación.

Eventos

VoiceXML proporciona un mecanismo de rellenado de formularios para el manejo de entradas de usuario "normales" y define un mecanismo para controlar los eventos no cubiertos por este procedimiento.

Los eventos son lanzados por la plataforma debido a una gran variedad de circunstancias, como por ejemplo, cuando el usuario no responde, no responde correctamente, solicita ayuda, etc. El intérprete también produce eventos si encuentra un error semántico en un documento de VoiceXML. El control de eventos comunes puede especificarse en cualquier nivel, y se aplica a todos los niveles inferiores.

- Enlaces: Los enlaces especifican transiciones a otros puntos del documento, otro documento dentro de la aplicación, u otro documento de otra aplicación.

2.5.2 Constructores de diálogo

Formularios

Nos referimos a los formularios como el componente fundamental de los documentos VoiceXML. Éstos contienen:

- Campos de entrada (ítems) y de control.
- Declaración de variables.
- Tratamiento de eventos.
- Acciones a ejecutar cuando se completen determinados campos.

Se definen dos tipos de formularios siguiendo el criterio del modo de interpretación:

Formularios directos: La estrategia se basa en ir recorriendo los campos de forma secuencial, eludiendo aquellos que se van completando adecuadamente.

La Figura 2-3 muestra un ejemplo de un formulario VoiceXML [17]. Para realizar la interpretación de los formularios se utiliza el algoritmo FIA (Form Interpretation Algorithm), que recorre el formulario almacenando el contenido de los campos, reproduce los textos asociados al formulario y comprueba si la condición <filled> (acciones a ejecutar cuando un determinado campo se completa) del formulario se cumple.

```
<form id="F1" scope="document">
<grammar scope="document" type="text/gsl">
  <![CDATA[
.MYRULE
    [[ (david hasselhoff) ] {<MySlot "dave">}}]]>
  </grammar>
<!-- the utterance of 'david hasselhoff' anywhere in the
      application will fill this namelist-->
<filled namelist="MySlot">
  <goto next="#F3"/> </filled>
<field name="F_1"> //Se define F1 como un campo de entrada
<grammar type="text/gsl">[(moe green)]</grammar>
  <prompt>
    Who should get the next crack at playing Hamlet ?
    His initials are d h .
  </prompt>
</field> <filled namelist="F_1"> //Se define lo que realiza el
sistema cuando rellena el campo F1
  <prompt>
    You said <value expr="F_1"/> Are you insane?
  </prompt>
  <goto next="#F2"/>
</filled>
</form>
```

Figura 0-3 - Ejemplo de formulario directo

Iniciativa mixta: Definiendo form-level grammars (gramáticas definidas externamente al formulario), las entradas se pueden completar en cualquier orden y se pueden rellenar varios campos en una única iteración. Además, pueden haber varias gramáticas correspondientes a formularios diferentes activas al mismo tiempo, determinando el sistema a qué campo del formulario se ha dado respuesta [17].

En un formulario pueden aparecer dos tipos de campos:

Ítems de entrada: Incorporan textos para informar al usuario sobre qué debe hacer, gramáticas que determinan si la entrada es válida o no y tratamiento de eventos. Se añade la acción <filled> para incorporar la acción a realizar cuando una variable de entrada se completa adecuadamente.

Ítems de control: Se trata de las etiquetas <block> (presenta información y no pide entrada de datos) e <initial> (bloque inicial del formulario para presentar información o ayuda sobre el mismo).

Cada campo del formulario tiene asociada una variable, que al final contendrá el resultado de interpretar el formulario y que se puede referenciar a lo largo de la aplicación.

Menús

Se trata del caso más simple de formulario, Sólo tiene una variable anónima la cual está obligada a elegir y esta se utilizará para decidir su siguiente transición.

Las opciones se especifican mediante la etiqueta <choice>, donde se indica una gramática de voz y/o de DTMF y un enlace a acudir cuando se selecciona la alternativa. El atributo accept se utiliza para determinar si el usuario debe indicar la opción completa (exact) o se acepten partes de la misma (approximate).

Respecto al modelo de interpretación, el funcionamiento es idéntico al de un formulario con un único campo que se encarga de realizar todo el trabajo, estableciéndose las indicaciones correspondientes al enlace mostrado en el objeto <choice>.

A continuación en la figura 2-4 se muestra un ejemplo de un Menú VoiceXML:

```
<menu id="M1" scope="dialog" dtmf="true">
  <prompt>
    Who is the most infamous celebrity in the weekly world
    news tabloid?
  </prompt>
  <prompt>
    <enumerate>
      For <value expr="_prompt"/>, press <value expr="_dtmf"/>
    </enumerate>
  </prompt>

  <choice event="bigfoot_event"> //Se especifican las opciones
    lie za manelly
  </choice>
  <choice event="bigfoot_event">
    bride of bigfoot
  </choice>
</menu>
```

Figura 0-4 - Ejemplo de Menú VXML

2.5.3 Gramáticas

El mecanismo elegido por VoiceXML para introducir un modelo de lenguaje se basa en la utilización de gramáticas, definidas mediante el objeto `<grammar>`. Una gramática determina la secuencia de palabras aceptables durante el proceso de reconocimiento automático del habla, y asimismo, incluye en el proceso de comprensión de las frases, pues indica la interpretación semántica de unidades sintácticas de las mismas.

En mayo de 2004 el W3C editó una serie de recomendaciones sobre el formato de gramáticas para el reconocimiento de voz denominadas Speech Recognition Grammar Specification [18]. Este documento puede verse como un complemento a las especificaciones de VoiceXML, tratando los casos de entradas de voz y entradas en formato DTMF.

En especificaciones SRGS se proponen dos formatos fundamentales de gramáticas:

- ABNF (Augmented BNF): Se trata de un formato en texto plano basada en formatos anteriores como el BNF (Backus Naur Form) y JSGF (Java Speech Grammar Format).
- XML: Se utiliza el formato XML, delimitando los diferentes campos entre las marcas correspondientes. Ambos formatos son independientes del contexto y son semánticamente equivalentes, desde el punto de vista que aceptan el mismo lenguaje como entrada y procesan cadenas de entrada de forma idéntica. En VoiceXML también se contemplan modelos de n-gramas, introducción de medidas de confianza y métodos de aplicación de suavizado como Back-Off, basado en el “rellenado” de la matriz de la gramática, sustituyendo los ceros de probabilidad existentes por no disponer de muestras de entrenamiento por valores adecuados.

Las gramáticas pueden ser introducidas de forma interna o externa. Las gramáticas internas se incluyen directamente en la etiqueta `<grammar>`. Las gramáticas externas se incluyen a través de URIs. El peso de las gramáticas se indica mediante el atributo `weight`, éste es indicado siguiendo el formato indicado en el SRGS. De éste modo se establece una jerarquía entre las diferentes gramáticas que conforman un formulario.

Las gramáticas internas a los campos de entrada sólo están activas cuando se ejecuta el campo correspondiente. Las gramáticas por enlaces poseen el alcance del elemento que engloba el link. Las gramáticas asociadas a formularios poseen el alcance del formulario, del documento o de la aplicación completa (caso de estar englobadas en el documento ROOT). Las gramáticas asociadas a menús poseen como alcance por defecto el del diálogo correspondiente, pero pueden darse los mismos casos que para las gramáticas del formulario.

Un formulario puede tener activas varias gramáticas de forma simultánea y desactivar otro conjunto. Cuando el intérprete permanece a la espera de una entrada tras visitar un campo, las gramáticas activas son:

- Gramáticas asociadas a dicho campo, inclusive la de los campos link.
- Gramáticas del formulario.
- Gramáticas contenidas en links dentro del documento.
- Gramáticas contenidas en el documento Root (enlaces, menús, formularios).
- Gramáticas definidas en la captura de eventos de la plataforma (ayuda, salida, cancelar).

Si existe más de una gramática activa del mismo nivel, la precedencia la marca el orden de los documentos. Como mínimo debe haber una gramática activa cuando se espera la respuesta de usuario. Se pueden desactivar gramáticas, de forma que simplemente éste vigente la del formulario en cuestión.

La interpretación semántica para el reconocimiento viene descrita en la especificación Semantic Interpretation for Speech Recognition [\[19\]](#). El resultado de la interpretación semántica debe trasladarse a variables ECMAScript, funcionalidad que debe ser soportada por el reconocedor. Cada campo de entrada tiene asociado un slot name, que almacena el resultado del reconocimiento para determinar si la entrada suministrada concuerda o no con la gramática. En caso de gramáticas asociadas a formularios, el resultado de reconocer adecuadamente uno o varios campos suponen la asignación de las variables ECMAScript y slots correspondientes. Las gramáticas asociadas a campos concretos sólo se activan cuando el algoritmo visita dicho campo.

CAPÍTULO 3. Sistema desarrollado

En este capítulo se describe la aplicación desarrollada para el Proyecto Final de Carrera. Se ha procurado hacer un sistema fácilmente extensible y totalmente reutilizable de manera que cualquier programador sin conocimientos de VXML pudiese crear formularios y menús nuevos.

3.1 Descripción de la infraestructura digital utilizada.

Se ha creado una infraestructura digital MVC con información de sesiones, de manera que esté separada la lógica de navegación, de la lógica de pintado y de los datos del modelo. Esto permite poder reutilizar en gran medida toda la lógica utilizada para crear vistas.

Esta infraestructura digital recibe las acciones a tomar por medio de los parámetros recibidos en la petición HTTP. De esta manera todas las peticiones son dirigidas a `index.php` con los dos siguientes parámetros: parámetro 'c' indicando el controlador a utilizar y parámetro 'a' indicando la acción que se va a utilizar.

Si el parámetro 'c' no estuviese asignado, se tomaría como controlador el controlador principal "IndexVxmlFilmController", el cual se detallará más adelante, a su vez si el parámetro 'a' no estuviese asignado se elegiría como acción por defecto la acción "index".

3.2 Lógica de navegación (Controladores)

La lógica de navegación se ha implementado por medio de un `NavigationMap` y `Controllers`. Estos sirven para decidir qué hará una determinada acción, si una petición HTTP es de un usuario identificado o anónimo y si tiene todos los datos necesarios.

Los flujos de navegación son definidos por enlaces en los `Controllers`, además de tener flujos por defecto como pueden ser por ejemplo la opción volver atrás (tecla *) y volver al menú principal (tecla 0) para simular la navegación de un Smartphone.

3.2.1 NavigationMap

El NavigationMap tiene la responsabilidad de decidir si una acción es válida o inválida. Para ello tiene definido todas las acciones del sistema en una array del siguiente tipo:

```
private static $CONFIG = array(  
    'Film' => array(  
        'getFilm' => array(  
            self::PARAMS => array('filmId', 'breadCrumb'),  
            self::OPTIONAL_PARAMS => array('breadCrumb'),  
        ),  
        'getFilmDetailed' => array(  
            self::PARAMS => array('filmId', 'breadCrumb'),  
            self::OPTIONAL_PARAMS => array('breadCrumb'),  
        ),  
        'getReview' => array(  
            self::PARAMS => array('filmId', 'breadCrumb', 'page', 'review_complete'),  
            self::OPTIONAL_PARAMS => array('breadCrumb'),  
            self::DEFAULT_VALUE => array('page' => 0, 'review_complete' => 0),  
        ),  
    ),  
    'Profile' => array(  

```

Figura 0-1 - Array Navigation Map

Ejemplo de configuración de NavigationMap

En este array se guarda por cada uno de los Controllers las acciones que se pueden ejecutar, los parámetros que necesitan esas acciones, cuáles de ellos son opcionales y cuáles de ellos tienen un valor por defecto en el caso de no estar definidos.

Además de los parámetros definidos en la configuración del NavigationMap, este también recoge la del Controller y la acción elegida en la petición además de la current_session y el current_caller_id los cuales hacen referencia a la sesión de la llamada y el identificador del llamador (Número de teléfono o ID de skype).

Con el parámetro de la sesión, el NavigationMap es capaz de decidir si es una sesión de usuario identificado o anónimo.

3.2.2 Controllers

Los Controllers son los encargados de hacer el control del flujo escogiendo información del modelo de las Apis y sistemas de almacenamiento y de crear vistas para presentar la información al usuario con la información y con las gramáticas necesarias para implementar los flujos de la aplicación en cada caso.

El Controlador general (Padre abstracto de los controladores) tiene la lógica para crear links pasándole el controlador del link, la acción y un array con los parámetros GET del link.

Todas las acciones reciben como parámetros un array con los parámetros de la petición y un texto que se inyectaría delante de la vista a utilizar. De esta manera se hace muy sencillo poder avisar al usuario de un error y redirigirle a otra página por medio de la inyección de textos.

Los Controladores están divididos por secciones, de manera que lo que se intenta tener es Controllers no demasiado grandes que se interconecten entre ellos.

La aplicación contiene los siguientes Controllers:

- IndexVxmlFilmController
- CarteleraController
- NextReleaseController
- LoginController
- ProfileController
- SearchController
- RecommendationController
- FilmController

Todos estos Controllers se detallan en los siguientes puntos de la memoria:

IndexVxmlFilmController

Este Controller es el encargado de mostrar el menú principal, tanto el menú principal para usuarios identificados como para usuarios anónimos.

Tiene una única acción por defecto (“index”), la cual dependiendo del estado de la sesión decide que menú elegir.

CarteleraController

Este controlador tiene como responsabilidad mostrar el menú referente a la cartelera y mostrar la cartelera en distintos órdenes.

Este orden puede ser el siguiente:

- Por fecha de estreno.
- Por puntuación en FilmAffinity
- Por número de votaciones en FilmAffinity (Numero de votaciones)

NextReleaseController

Este controlador tiene como responsabilidad mostrar el menú referente a los próximos estrenos y mostrar los próximos estrenos en distintos órdenes.

Este orden puede ser el siguiente:

- Por fecha de estreno.
- Por puntuación en FilmAffinity
- Por número de votaciones en FilmAffinity (Numero de votaciones)

LoginController

Este controlador tiene como responsabilidad el proceso de identificación y registro de usuario.

Para el proceso de identificación, se pedirá el usuario y la contraseña, guardando previamente el usuario en la sesión para que nunca fuese necesario mandar el usuario y la contraseña en la misma petición, ya que sería peligroso.

El LoginController además también comprueba si el currentCallerId (desde donde estas llamando) ya está asociado a algún usuario, y si es así comprobar si tiene la contraseña para ese teléfono, y poder iniciar la identificación por teléfono con o sin contraseña, dependiendo si el usuario tuviese la contraseña guardada o no.

Para el registro se le pedirá al usuario su número de teléfono, su contraseña (8 dígitos) y que lo repita de nuevo.

Una vez finalizado el registro, el usuario será redirigido al ProfileController para que gestione sus preferencias.

ProfileController

El ProfileController tiene la responsabilidad de gestionar el perfil del usuario. Este controlador solo es accesible para peticiones con la sesión identificada, si la sesión no estuviese identificada, se redirigiría al menú principal con un mensaje de error.

Dentro del ProfileController se pueden gestionar las distintas preferencias del usuario:

- Gestionar provincia del usuario
- Gestionar cines favoritos del usuario (Necesario tener provincia seleccionada)
- Gestionar horario preferido para ir al cine
- Gestionar directores favoritos
- Gestionar lista negra de directores
- Gestionar actores favoritos
- Gestionar lista negra de actores
- Gestionar géneros de película favoritos
- Gestionar lista negra de géneros de película

SearchController

El SearchController tiene la responsabilidad de mostrar el menú de búsquedas y los resultados de búsquedas para los siguientes tipos de búsquedas:

- Búsqueda por título.
- Búsqueda por director.
- Búsqueda por actor.

RecommendationController

El RecommendationController tiene la responsabilidad de mostrar al usuario lo que puede ir a ver al cine dependiendo de sus preferencias.

Este Controller solo puede ser usado por sesiones identificadas de manera que si no está identificado será redirigido al menú principal con un mensaje de error. Si la sesión está identificada, el usuario dueño de la sesión deberá de tener configurado sus cines favoritos, si no será redirigido al ProfileController con un mensaje de error.

FilmController

El FilmController tiene la responsabilidad de mostrar la información de películas. Este Controller puede ejecutar las siguientes acciones:

- Mostrar listados de películas paginados
- Mostrar información de una película
- Mostrar información detallada de una película
- Mostrar las críticas de una película ocultando los spoilers de forma paginada.
- Mostrar una crítica completa.

Diagrama de clases de Controllers

En la Figura 3-2 podemos observar el diagrama de clases de todas los Controllers:

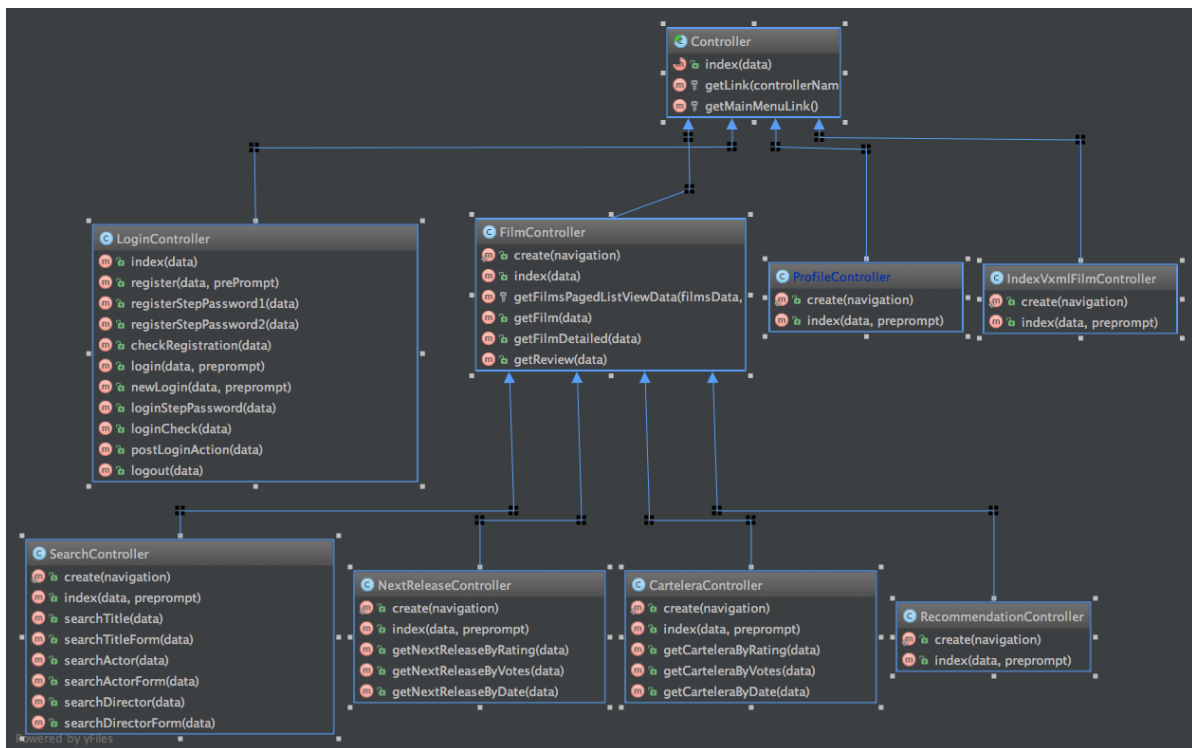


Figura 0-2 - Diagrama de clases de Controllers

3.3 Lógica de pintado (Vistas)

La lógica de pintado está dividida en tres tipos de clases: View, ViewData y Template.

- Las View comprueban que la ViewData sea correcta y eligen el Template a utilizar.
- Las ViewData tienen la información necesitada por el Template para poder ser pintado.
- Los Templates son plantillas dinámicas con código VXML las cuales son rellenas por la información de la ViewData.

Además de estos tres tipos de clases hay unas cuantas clases en las que se apoyan, para transformar el lenguaje normal a lenguaje VXML, como puede ser transformar números a pulsaciones de teléfono (DTMF) o verificar que los nombres que el sistema va a dictar o reconocer van a ser válidos.

3.3.1 Templates

Actualmente solo hay dos Templates. Estos Templates son plantillas PHTML de VXML en los cuales se rellenan campos como el texto, la gramática y la información a enviar al sistema.

Los dos Templates implementan un formulario VXML, como hemos dicho anteriormente la finalidad de la infraestructura digital es hacer lo más sencillo posible al programador crear nuevos flujos VXML. Por esto todas las acciones son formularios VXML que encadenan otra acción.

Estos dos Templates son Menú y Form, los cuales especificaremos a continuación.

Template Menú

Este Template implementa un formulario con opciones cerradas por lo que cada una de las opciones definidas en la gramática corresponde a un enlace distinto (puede que varias opciones correspondan al mismo enlace). Por lo que una vez que la gramática identifica una opción válida dicha por el usuario, en el Template se elige el enlace a utilizar para realizar una petición al sistema.

Además de las opciones especificadas por la ViewData para implementar el menú, también son añadidas las opciones básicas de navegación, como volver a atrás e ir al menú principal, si estos enlaces están definidos en la ViewData.

Tanto el texto, el título, la gramática y los enlaces a usar con cada una de las opciones son dados por la ViewData.

Todos los enlaces a los que se accedan se harán por peticiones HTTP POST y se añadirán además de los parámetros necesarios para la acción, los parámetros ocultos definidos en la ViewData. Estos parámetros ocultos son definidos como variables en el Template.

Este Template es usado por páginas como el menú principal, en la que se le da una serie de opciones al usuario para navegar por la aplicación.

Template Form

Este Template implementa un formulario con opciones abiertas. Por lo que todas las opciones definidas por la gramática ejecutan la misma acción a la que se le envía la opción reconocida por la gramática como parámetro.

Además de la acción especificada en la ViewData, también son añadidas las opciones básicas de navegación, volver atrás e ir al menú principal, si estos enlaces están definidos en la ViewData.

Todos los enlaces a los que se acceda se harán por peticiones HTTP POST y además se añadirán los parámetros necesarios para la acción, los parámetros ocultos definidos en la ViewData. Estos parámetros ocultos son definidos como variables en el Template.

Este Template es usado por páginas en el que el usuario debe de decir una opción cuando el sistema no le da una lista de opciones. Por ejemplo, en el formulario de búsqueda por director en el que el usuario ha de decir el nombre del director a buscar.

3.3.2 ViewData

Las ViewData son contenedores de datos, que con el uso de varias clases de apoyo adecuan la información para rellenar la información necesitada por los Templates.

Además, las ViewDatas son capaces de generar las gramáticas necesarias a partir de la información insertada desde los Controllers.

El sistema desarrollado contiene las siguientes ViewData (en los siguientes puntos de la memoria se detallara la funcionalidad de cada una de ellas):

- SimpleViewData
- MenuViewData
- FormViewData
- PagedListViewData
- FilmBasicViewData
- FilmDetailedViewData
- ReviewViewData
- ShowTimeViewData

SimpleViewData

Esta es la ViewData padre abstracta. Tiene la funcionalidad básica para ser una ViewData.

En el SimpleViewData están definidos los métodos para gestionar las opciones básicas de navegación, volver atrás e ir al menú principal, así como la gestión de los parámetros ocultos enviados a la acción.

Los parámetros ocultos se usan para poder mandar en cada acción por HTTP POST, cual es la sesión actual del usuario, migas de pan (breadcrumbs) y cuál es el actual ID del llamador (teléfono o cuenta de skype desde donde está llamando).

Esta ViewData tiene un método abstracto que es el “*getGrammar*”, este método se ocupa de dar la gramática VXML para poder reconocer las respuestas del usuario.

Todas las demás ViewDatas son hijas directa o indirectamente de esta SimpleViewData.

MenuViewData

Esta ViewData tiene la responsabilidad de ser el contenedor de datos para los menús simples. Estos son los menús que tienen las opciones pasadas por el Controller y no auto añade ninguna opción, salvo las opciones de navegación básica.

Las opciones son añadidas dando información de la opción, la frase o tecla para seleccionar la opción y el enlace con la acción a la que te llevaría esta opción.

Para esta ViewData es necesario rellenar el texto y las opciones.

Esta ViewData genera automáticamente la gramática VXML dependiendo de las opciones introducidas, de esta manera si desde el controlador se decidiese añadir una nueva opción solo habría que modificar el texto si se quiere y añadir la opción. Entre el Template y la MenuViewData se reflejaría la opción necesaria sin añadir ni una sola línea de VXML.

La gramática generada por la ViewData será de tipo GSL y se define de la siguiente manera:

```
<grammar type="text/gsl">
  [[(buscar          películas)(dtmf-1)(uno)(1)(cartelera)(dtmf-2)(dos)(2)(próximos
estrenos)(dtmf-3)(tres)(3)(loguin)(dtmf-4)(cuatro)(4)(registrar)(dtmf-5)(cinco)(5)] (dtmf-
0)(dtmf-star)]
</grammar>
```

Tabla 0-1 - Gramática generada por la ViewData

En este caso la ViewData habría generado una gramática con 5 opciones, la cual se pueden elegir diciendo una frase, tecleando un número o diciendo el número, además de las opciones básicas que serían pulsar la tecla 0 y la tecla *.

Para la generación de la gramática la ViewData se ayuda de una clase de apoyo llamada MenuOption, la cual se verá más adelante, que se ocupa de escapar y quitar símbolos que VXML no acepte como palabras acentuadas.

Form ViewData

Esta ViewData tiene la responsabilidad de ser el contenedor de datos para los formularios. Estos son las páginas que tienen una respuesta abierta por parte del usuario y una vez reconocida esta respuesta por la gramática, esta respuesta es pasada como parámetro a la acción.

Para esta ViewData es necesario definir su texto, el enlace de la acción, el nombre de la variable donde se va a guardar la respuesta y la gramática para reconocer la respuesta del usuario.

La FormViewData generará dos gramáticas, una para la navegación básica y otra para la selección de opción. De esta manera la gramática generada para la navegación básica solo aceptará pulsaciones de teclado, mientras que la gramática para reconocer la respuesta del usuario será una gramática multilenguaje. Ya que para buscar películas, nombres de actores y directores no tiene sentido hacer intentar pronunciar en castellano nombres en inglés.

La gramática para reconocer la respuesta del usuario puede ser introducida de tres maneras:

- Gramática externa.
- Gramática generada desde CSV.
- Gramática generada desde opciones.
- Gramática generada para lista de dígitos

Gramática externa.

A la ViewData se le indicará la ruta donde se encuentra el fichero que contenga la gramática. En esta aplicación no se ha añadido ninguna así. Ya que esta opción implica escribir más código VXML y la finalidad de esta infraestructura digital es evitar eso.

Gramática generada desde CSV

La gramática generada desde CSV se generará pasándole a la ViewData la ruta del CSV y lo convertirá en opciones.

Este CSV tendrá el siguiente formato:

Locale_1, opcion1, opcion2...
Locale_2, opcion3, opcion4....

Teniendo por Locale_1 y Locale_2 los distintos lenguajes válidos para la aplicación (de momento español de España e inglés)

Los CSVs están alojados en la siguiente ruta src/grammar/csv/XXX.csv y en un futuro se podría separar del proyecto para tenerlos en un repositorio de configuración.

En la siguiente tabla se puede observar un ejemplo del CSV utilizado para reconocer nombres de directores:

es-ES,Alex de la Iglesia, Alejandro Amenábar, Julio Medem, en-US,Cohen,Clint EastWood, Alfred Hitchcock, Steven Spielberg, Ridley Scott, James Cameron
--

Tabla 0-2 - Ejemplo del CSV utilizado para reconocer nombres de directores

Gramática generada desde opciones

Pueden ser opciones generadas desde el código, introduciéndolas una a una u opciones generadas desde un CSV.

Estas opciones, en el caso de ser una frase compuesta de varias palabras, generará la opción para la frase compuesta y la opción para cada una de las palabras si se desea, siempre y cuando esta opción no esté ya creada o sea una opción restringida (opciones de navegación básica). En el caso de esta aplicación se ha configurado para que separe las opciones en sub-opciones, ya que las respuestas de los formularios son títulos de películas y nombres de personas.

La gramática está configurada para ser una gramática recursiva, por lo que reconocerá de 1-n respuestas válidas, esto hace que al introducir varias frases compuestas la gramática será capaz de reconocer nuevas frases creadas a partir de las frases anteriores, incluso frases multilenguaje en las que una parte sea en castellano y otra en inglés (como por ejemplo algunos nombres de actores).

A continuación, se indica un ejemplo de la gramática creada para la búsqueda de directores:

<pre><grammar mode="voice" root="voiceRequest"> <rule id="voice_inputs"> <one-of> <item xml:lang="es-ES">alex</item> <item xml:lang="es-ES">de</item></pre>

```
<item xml:lang="es-ES">la</item>
<item xml:lang="es-ES">iglesia</item>
<item xml:lang="es-ES">alex de la iglesia</item>
<item xml:lang="es-ES">alejandro</item>
<item xml:lang="es-ES">amenabar</item>
<item xml:lang="es-ES">alejandro amenabar</item>
<item xml:lang="es-ES">julio</item>
<item xml:lang="es-ES">medem</item>
<item xml:lang="es-ES">julio medem</item>
<item xml:lang="en-US">cohen</item>
<item xml:lang="en-US">clint</item>
<item xml:lang="en-US">eastwood</item>
<item xml:lang="en-US">clint eastwood</item>
<item xml:lang="en-US">alfred</item>
<item xml:lang="en-US">hitchcock</item>
<item xml:lang="en-US">alfred hitchcock</item>
<item xml:lang="en-US">steven</item>
<item xml:lang="en-US">spilberg</item>
<item xml:lang="en-US">steven spilberg</item>
<item xml:lang="en-US">ridley</item>
<item xml:lang="en-US">scott</item>
<item xml:lang="en-US">ridley scott</item>
<item xml:lang="en-US">james</item>
<item xml:lang="en-US">cameron</item>
<item xml:lang="en-US">james cameron</item>
</one-of>
</rule><rule id="voiceRequest" scope="public">
<one-of>
<item repeat="1-"> <ruleref uri="#voice_inputs"/> </item>
</one-of>
</rule>
</grammar>
<grammar mode="dtmf" root="request">
<rule id="navigation">
<one-of>
<item>0</item>
<item>*</item>
</one-of>
</rule><rule id="request" scope="public">
<one-of>
<item> <ruleref uri="#navigation"/> </item>
</one-of>
</rule>
</grammar>
```

Tabla 0-3 - Ejemplo de gramática creada para la búsqueda de directores

Gramática generada para una lista de dígitos

Si la respuesta esperada en un formulario es una lista de dígitos, se pueden añadir opciones numéricas especificando la longitud de la lista de maneras.

De esta manera, si se espera un teléfono (sin código de país) se añadirá un input numérico de 9 dígitos.

En el siguiente ejemplo, se puede observar la gramática autogenerada para el formulario de identificación:

```
<grammar mode="dtmf" root="request">
<rule id="navigation">
<one-of>
<item>0</item>
<item>*</item>
</one-of>
</rule><rule id="allDigits">
<one-of>
<item>0</item>
<item>1</item>
<item>2</item>
<item>3</item>
<item>4</item>
<item>5</item>
<item>6</item>
<item>7</item>
<item>8</item>
<item>9</item>
</one-of>
</rule><rule id="request" scope="public">
<one-of>
<item> <ruleref uri="#navigation"/> </item>
<item repeat="9"> <ruleref uri="#allDigits"/> </item>
</one-of>
</rule>
</grammar>
```

Tabla 0-4 - Gramática autogenerada para el formulario de identificación

PagedListViewData

Esta ViewData tiene la responsabilidad de mostrar un menú de elementos paginados.

Es usada para mostrar listas de búsquedas de películas, de carteleras y listas de cines. Hereda la lógica de MenuViewData, por lo que es necesario definir las opciones de la página que se está mostrando en el momento, además es necesario añadir el número total de páginas, el número de la página actual y los links en el caso de que existan para la página inicial, pagina anterior, pagina siguiente y última página, para crear la paginación.

No es necesario definirle el texto ya que lo autogenera con la información de cada una de las opciones, más la información de la paginación y la información de navegación básica.

Para la paginación, se utiliza por defecto la palabra página, así que por defecto se referirá a todas las opciones de paginación como: página siguiente, página anterior... Pero puede ser definido con el nombre que se quiera. De hecho, más adelante veremos ejemplos de ViewData que heredan la lógica de PagedListViewData, las cuales sobrescribirán el nombre de la paginación.

La gramática generada por esta ViewData, es igual que la generada por su padre MenuViewData inyectando las opciones para la paginación.

FilmBasicViewData

Hasta ahora todas las ViewData vistas eran “ViewData generales” que podían ser usadas por cualquier modelo de datos y por cualquier aplicación. La diferenciación se las da el controlador al definirle el texto y las opciones que se quieren dar al usuario para interactuar. Las siguientes ViewData que vamos a describir son especificaciones ajustadas al modelo.

La FilmBasicViewData, es una ViewData que hereda la lógica de MenuViewData que contiene la información básica de una película.

Es necesario definir la película de clase Film que se quiere mostrar, el link a la información detallada de dicha película y el link a las críticas de usuario de la película además de la navegación básica.

Esta ViewData generará un texto con la información básica de la película y una gramática con la opción de escuchar más detalles de la película y escuchar las críticas de usuario.

FilmDetailedViewData

La FilmDetailedViewData es una ViewData que hereda la lógica de MenuViewData y contendrá la información detallada de una película.

Es necesario definir la película de clase Film que se quiere mostrar y el link a las críticas de usuario.

Esta ViewData generará un texto con la información detallada de la película y una gramática con las opciones para escuchar las críticas de usuario, además de la navegación básica.

ReviewViewData

La ReviewViewData tiene la responsabilidad de contener los datos necesarios para mostrar la información de una crítica de usuario.

Hereda toda la lógica de PagedListViewData, por lo que muestra las críticas de forma paginada, con la peculiaridad de que nos las muestra de 1 en 1. Por ello sobrescribe el nombre de la paginación de página a crítica.

Esta ViewData se puede configurar para que muestre una crítica con la parte que contenga spoilers escondida. Añadiendo la opción para escuchar la crítica entera.

Para esta ViewData es necesario definir la crítica que se quiere mostrar, si se quiere mostrar la crítica completa (con spoilers) o no, los links para la paginación y el link para mostrar la crítica completa en el caso de que se estén ocultando los spoilers.

El texto es auto generado con la crítica más la información de paginación más la información de navegación básica.

ShowTimeViewData

La ShowTimeViewData tiene la responsabilidad de contener los datos necesarios para mostrar la información de las sesiones de una película.

Hereda toda la lógica de PagedListViewData y también muestra solo una película por página por lo que cambia el nombre de la paginación de página a película.

Para esta ViewData es necesario definir el ShowTime (sesiones de una película) que se quiere mostrar y el link a la información de la película, además de los links para la paginación.

El texto es auto generado con la crítica, con la información de paginación y con la información de navegación.

Diagrama de clases de ViewData

Como se ha comentado en los puntos anteriores, las ViewData son contenedores de datos que con el uso de varias clases de apoyo adecuan la información para rellenar la información necesitada por los Templates y son capaces de generar las gramáticas necesarias a partir de la información insertada desde los Controllers.

En la siguiente figura podemos observar el diagrama de clase de todas las ViewDatas:

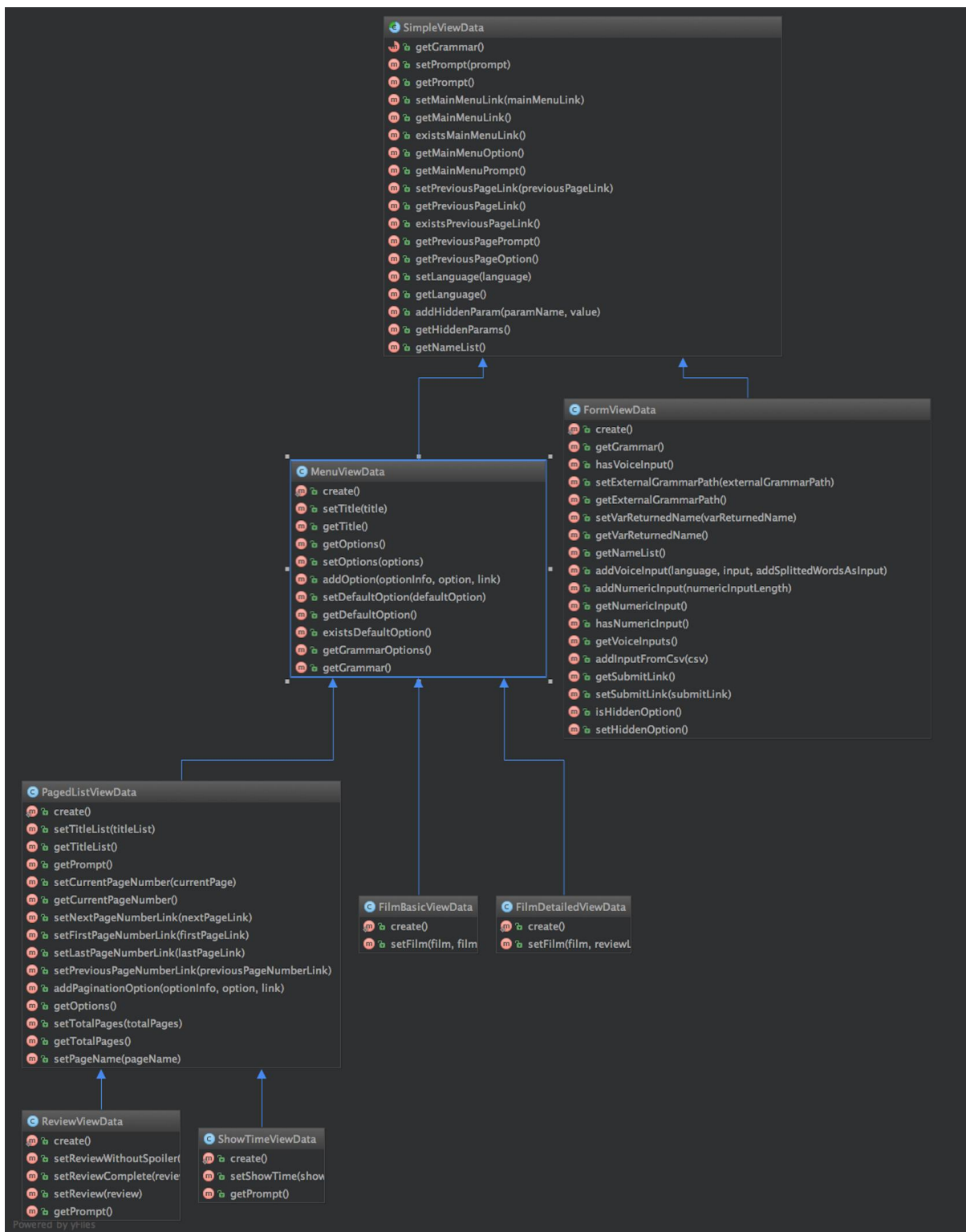


Figura 0-3 - Diagrama de clases de ViewData

Las Views son las encargadas de recoger los datos de la ViewData, validar que pueden ser mostrados correctamente y añadirlos al Template que tengan definido.

Actualmente existen 2 tipos de View:

- FormView
- MenuView

FormView

Recoge una ViewData de tipo FormViewData y la añade al Template “Form.phtml”. Además, comprueba que la ViewData tenga un texto definido y que contenga una gramática valida, ya sea porque la gramática ha sido añadida como gramática externa o porque la gramática es autogenerada y por lo tanto, tiene alguna opción para identificar la respuesta del usuario, ya sea una opción de voz o numérica.

MenuView

Recoge una ViewData de tipo MenuViewData o alguno de sus subtipos y la añade al Template “Menu.phtml”.

Además, comprueba que la ViewData tenga un texto definido y alguna opción válida para generar la gramática, aunque sea una opción de navegación básica

Diagrama de clases de Views

En la siguiente figura podemos observar el diagrama de clases de las Views descritas en los puntos anteriores:

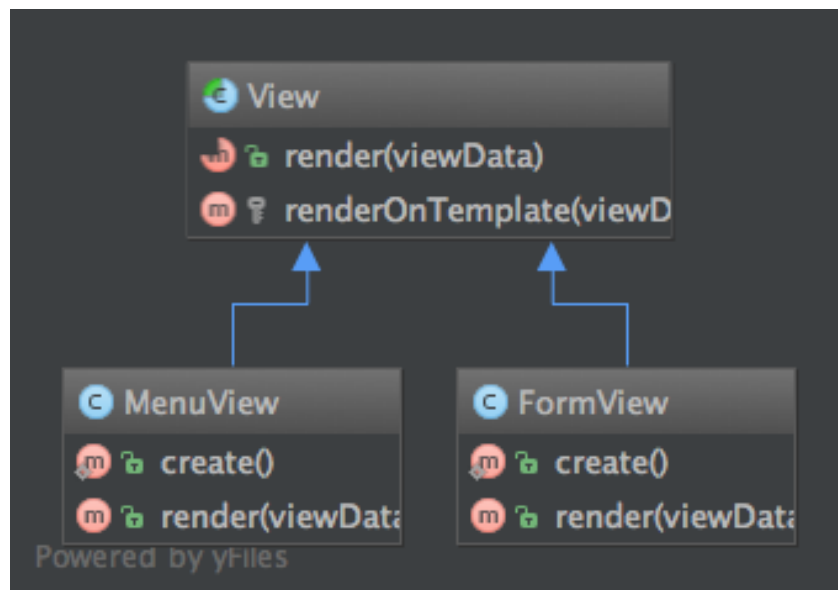


Figura 0-4 - Diagrama de clases de Views

3.3.3 Clases de Apoyo

Tanto las ViewData como los Template se apoyan en varias clases de apoyo:

- InputSanitizer
- KeyPhone
- Language
- Link
- FormInput
- MenuOption

InputSanitizer

La clase InputSanitizer tiene la responsabilidad de escapar una entrada numérica y eliminar los espacios que hay entre ellos.

Esta clase es necesaria ya que los inputs numéricos de los formularios del sistema los devuelve como números separados por espacios.

KeyPhone

La clase KeyPhone tiene la responsabilidad de devolver los tonos de teclados válidos para las gramáticas, devolver su valor numérico o su nombre a través de su valor DTMF y viceversa.

Language

La clase Language tiene la responsabilidad de albergar los lenguajes válidos para la aplicación, así como de validar si un lenguaje es válido o no.

Link

La clase Link tiene la responsabilidad de encapsular la lógica de crear links.

De esta manera es creado con el Controlador al que se quiere acceder, la acción que se quiere ejecutar en el Link y una array de parámetros que se pasaran como parámetros GET. Este array estará definido como una Hash Map, en el cual la clave será el nombre del parámetro y el valor de este.

El enlace puede devolver la URL con los valores escapados para poder ser mostrada en VXML, como URL a la que acceder o como URL que pasar como variable.

FormInput

La clase FormInput tiene la responsabilidad de representar una respuesta o parte de una respuesta válida para la gramática de un formulario (formulario con respuesta abierta).

Este tiene la información del idioma en la que está el input y es capaz de eliminar caracteres inválidos para las gramáticas VXML.

MenuOption

La clase MenuOption tiene la responsabilidad de representar una opción válida para un menú.

El MenuOption tiene la responsabilidad de guardar una opción válida del menú para el usuario, esta opción contiene la respuesta del usuario para elegirla, su descripción y el Link que contiene la transición que realizará la opción.

El MenuOption es capaz de eliminar y validar las opciones, de manera que no contengan caracteres inválidos para las gramáticas VXML.

3.4 Modelo (Apis y Sistemas de almacenamiento)

El Modelo del sistema se puede diferenciar en tres tipos:

- Modelo de datos externo.
- Modelo de datos locales.
- Modelo de datos mixto.

3.4.1 Modelo de datos externo

Con el modelo de datos externo nos referimos a los datos de la aplicación que no son albergados por nosotros. Como por ejemplo toda la información de FilmAffinity.

Para recoger la información de FilmAffinity se hace por medio de una técnica de extracción de datos de sitios web (web scraping).

Para ello se ha utilizado la librería <http://sourceforge.net/projects/simplehtmldom/>, la cual es OpenSource y sirve para poder navegar en el DOM del HTML.

Esto es utilizado a través de FilmAffinityApi.

FilmAffinityApi

Modelo de datos

Esta Api genera peticiones hacia FilmAffinity para recoger información de su web, analiza el HTML obtenido y lo devuelve encapsulado en las siguientes clases.

- Cinema
- Film
- Review
- SessionTime
- ShowTime

Cinema

Representa un cine, contiene el ID del cine en FilmAffinity, la dirección y el nombre del cine.

Film

Representa la ficha de una película.

La información que se mostrará sobre la película es la siguiente:

- ID de película.
- Título.
- Título original.
- Año.
- Duración.
- País.
- Director.
- Casting (Actores).
- Género de la película.
- Guionista.
- Compositor/es banda sonora.
- Responsables de fotografía.
- Productor.
- Sinopsis.
- Críticas profesionales.
- Numero de Críticas de los usuarios.
- Fecha de estreno (Opcional).
- Puntuación FilmAffinity (Opcional).
- Número de votos (Opcional).
- Puntuación en base a los gustos del usuario (Opcional).

Review

Representa una crítica sobre una película hecha por un usuario de FilmAffinity.
La información que se mostrará sobre la Review es la siguiente:

- Título.
- Autor.
- Fecha de creación.
- Parte de crítica libre de spoilers (contenido que desvela el argumento de la película).
- Parte de crítica con spoilers (contenido que desvela el argumento de la película)

SessionTime

Representa las sesiones emitidas por un cine para ver una película.
La información que se mostrará sobre las sesiones de un cine es la siguiente:

- Cinema.
- Lista de tiempos de las sesiones.

ShowTime

Representa las sesiones en las que se proyecta una película.
La información que se mostrará sobre la proyección de la película es la siguiente:

- Film.
- Lista de SessionTime.

Funcionalidad

Esta API hace peticiones a FilmAffinity y analiza el HTML obtenido para recoger la información necesaria de las webs.
La FilmAffinityApi ofrece la siguiente funcionalidad:

- `getReview($filmId, $page)`
- `getShowtimes($provinceId, $cinemaIds, $startHour, $endHour)`
- `getCinemasPaginated($provinceId, $page, $cinemasPerPage)`
- `getCinemas($provinceId, array $cinemaIds)`
- `getFilm($filmId)`
- `searchDirector($directorName, $page = 0, $filmsPerPage = 9)`
- `searchTitle($title, $page = 0, $filmsPerPage = 9)`
- `searchActor($actorName, $page = 0, $filmsPerPage = 9)`
- `getCarteleraRatingSorted($pageNumber, $filmsPerPage)`
- `getCarteleraVotesSorted($pageNumber, $filmsPerPage)`

- `getCarteleraReleaseDateSorted($pageNumber, $filmsPerPage)`
- `getNextReleaseRatingSorted($pageNumber, $filmsPerPage)`
- `getNextReleaseVotesSorted($pageNumber, $filmsPerPage)`
- `getNextReleaseReleaseDateSorted($pageNumber, $filmsPerPage)`

`getReview`

Función que devuelve una Review (crítica) de la película (filmId) que se encuentre dentro de la página pasada por argumento.

Ambos parámetro son pasados por argumento y este método devolverá una página por crítica.

Parámetros:

- `$filmId` (identificador de la película)
- `$page` (página de crítica que se quiere obtener)

Retorno:

- Review (la crítica encontrada)

`getShowtimes($provinceId, $cinemaIds, $startHour, $endHour)`

En esta función se pasan como argumentos los cines, provincias y las horas solicitadas para la proyección de una película y se devuelve una lista de ShowTimes (proyecciones de películas) con las preferencias elegidas por el usuario.

Estos ShowTimes estarán ordenados por puntuación.

Parámetros:

- `$provinceId` (identificador de la provincia de la búsqueda)
- `$cinemaIds` (lista de los identificadores de cines donde se quiere buscar)]
- `$startHour` (mínima hora a la que debe de empezar la película)
- `$endHour` (máxima hora a la que debe de acabar la película)

Retorno:

- ShowTimes[] (lista de proyecciones de películas)

`getCinemasPaginated($provinceId, $page, $cinemasPerPage)`

Función que devuelve una lista de cines de forma paginada.

Parámetros:

- `$provinceId` (identificador de la provincia donde se quiere buscar)
- `$page` (número de página que se quiere consultar)
- `$cinemasPerPage` (número de cines por página que se quieren en la respuesta)

Retorno (Array):

- Entero (Número total de páginas)
- Cinema[] (lista de cines)

`getCinemas($provinceId, $cinemaIds)`

Función que devuelve una lista de cines buscados por los identificadores de cines.

Parámetros:

- \$provinceId (identificador de la provincia donde se quiere buscar)
- \$cinemaIds (lista de identificadores de los cines que se quieren consultar)

Retorno:

- Cinema[] (lista de cines)

`getFilm($filmId)`

Función que devuelve una película por Identificador

Parámetros:

- \$filmId (identificador de la película que se quiere buscar)

Retorno:

- Film (película)

`searchDirector($directorName, $page = 0, $filmsByPage = 9)`

Función que realiza una búsqueda por director.

Parámetros:

- \$directorName(nombre del director a buscar)
- \$page (Número de página que se quiere consultar)
- \$filmsByPage (Número de películas por página)

Retorno (Array):

- Entero (Número total de páginas)
- Array(filmId => titulo) (lista de títulos indizados con identificadores de película)

`searchActor($actorName, $page = 0, $filmsByPage = 9)`

Función que realiza una búsqueda por actor.

Parámetros:

- \$actorName(nombre del actor a buscar)

- \$page (Número de página que se quiere consultar)
- \$filmsByPage (Número de películas por página)

Retorno (Array):

- Entero (Número total de páginas)
- Array(filmId => titulo) (lista de títulos indizados con identificadores de películas)

`searchTitle($title, $page = 0, $filmsByPage = 9)`

Función que realiza una búsqueda por título.

Parámetros:

- \$title(título a buscar)
- \$page (Número de página que se quiere consultar)
- \$filmsByPage (Número de películas por página)

Retorno (Array):

- Entero (Número total de páginas)
- Array(filmId => titulo) (lista de títulos indizados con identificadores de películas)

`getCarteleraRatingSorted ($page = 0, $filmsByPage = 9)`

Función que devuelve la cartelera ordenada por puntuación.

Parámetros:

- \$page (Número de página que se quiere consultar)
- \$filmsByPage (Número de películas por página)

Retorno (Array):

- Entero (Número total de páginas)
- Array(filmId => titulo) (lista de títulos indizados con identificadores de películas)

`getCarteleraVotesSorted ($page = 0, $filmsByPage = 9)`

Función que devuelve la cartelera ordenada por número de votos.

Parámetros:

- \$page (Número de página que se quiere consultar)
- \$filmsByPage (Número de películas por página)

Retorno (Array):

- Entero (Número total de páginas)
- Array(filmId => titulo) (lista de títulos indizados con identificadores de películas)

`getCarteleraReleaseDateSorted ($page = 0, $filmsByPage = 9)`

Función que devuelve la cartelera ordenada por fecha de estreno.

Parámetros:

- \$page (Número de página que se quiere consultar)
- \$filmsByPage (Número de películas por página)

Retorno (Array):

- Entero (Número total de páginas)
- Array (filmId => titulo) (lista de títulos indizados con identificadores de películas)

`getNextReleaseRatingSorted ($page = 0, $filmsByPage = 9)`

Función que devuelve los próximos estrenos ordenados por puntuación.

Parámetros:

- \$page (Número de página que se quiere consultar)
- \$filmsByPage (Número de películas por página)

Retorno (Array):

- Entero (Número total de páginas)
- Array (filmId => titulo) (lista de títulos indizados con identificadores de películas)

`getNextReleaseVotesSorted ($page = 0, $filmsByPage = 9)`

Función que devuelve los próximos estrenos ordenados por número de votos.

Parámetros:

- \$page (Número de página que se quiere consultar)
- \$filmsByPage (Número de películas por página)

Retorno (Array):

- Entero (Número total de páginas)
- Array (filmId => titulo) (lista de títulos indizados con identificadores de películas)

`getNextReleaseReleaseDateSorted ($page = 0, $filmsByPage = 9)`

Función que devuelve los próximos estrenos ordenados por fecha de estreno.

Parámetros:

- \$page (Número de página que se quiere consultar)
- \$filmsByPage (Número de películas por página)

Retorno (Array):

- Entero (Número total de páginas)
- Array(filmId => titulo) (lista de títulos indizados con identificadores de películas)

3.4.2 Modelo de datos locales

Se almacena información local en base de datos, la base de datos es accedida por un conjunto de clases llamados DBStorage que representan las tablas y el acceso a MySQL y por una clase DBStorageObject que representan una fila de cada tabla.

DbStorage

El DbStorage es la clase padre con toda la lógica de acceso a base de datos, excepto la información de conexión de base de datos que se encuentra en una clase llamada DBConfig, la cual podría ser sacada a otro repositorio con solo configuraciones del proyecto.

Todas las consultas que ejecuta el DBStorage son parametrizadas de manera para cubrirnos ante ataques de mysqlInjection (inyección de sentencias MySQL).

UserStorage

El UserStorage se encarga de almacenar la información básica del usuario.

En la siguiente tabla se pueden observar los campos que se van a almacenar de los usuarios, sus tipos y sus descripciones:

Columna	Tipo	Descripción
Pone	INT UNSIGNED NOT NULL Primary key	Teléfono del usuario
password	INT UNSIGNED	Contraseña del usuario
province_id	VARCHAR(32)	Identificador de provincia
start_favourite_schedule	TINYINT	Inicio de su hora favorita para ir al cine
end_favourite_schedule	TINYINT	Fin de su hora favorita para ir al cine
register_status	TINYINT	Estado del usuario (0 registrándose, 1 registrado)
register_time	INT UNSIGNED	Fecha de registro, timestamp

Tabla 0-5 - Información básica del usuario

SessionStorage

El SessionStorage almacena la información de una sesión. Es utilizado para saber si una sesión está identificada o está registrándose... etc.

En la siguiente tabla se pueden observar los campos que se van a almacenar de la información de la sesión, sus tipos y sus descripciones:

Nombre	Tipo	Descripción
phone	INT UNSIGNED NOT NULL Primary key	Teléfono del usuario
session_id	VARCHAR(64) PRIMARY_KEY	Identificador de la sesión
session_time	INT UNSIGNED	Fecha de la expiración de la sesión
session_status	TINYINT	El estado de la sesión (0 Anónimo, 1 Registrando, 2 En proceso de identificación, 3 Identificado)

Tabla 0-6 - Información de una sesión

CallerStorage

El CallerStorage almacena información sobre los teléfonos (identificadores de dispositivos) que se utilizan para llamar. Es utilizado para saber si un dispositivo está asociado a algún usuario y si la contraseña esta almacenada para ese dispositivo. De esta manera se puede hacer auto identificación. En la siguiente tabla se pueden observar los campos que se van a almacenar de los teléfonos, sus tipos y sus descripciones:

Nombre	Tipo	Descripción
caller_id	VARCHAR(64) NOT NULL PrimaryKey	Identificador del terminal con el que se llama.
phone	INT UNSIGNED	Teléfono del usuario que llamó con este dispositivo
first_identificación_time	INT UNSIGNED	Primera vez que se identificó con este terminal
last_identificación_time	INT UNSIGNED	Ultima vez que se identificó con este terminal
remember_flag	TINYINT	Marca para recordar contraseña (0 no recordar, 1 recordar)

Tabla 0-7 - Información sobre los teléfonos

FilmPreferencesStorage

Esta clase se usa para almacenar las preferencias del usuario. Los datos se almacenan en tipo BLOB en formato JSON, ya que nunca se buscará por estos campos. En un futuro si fuese necesario se podrían crear más tablas y normalizar las tablas.

Esta clase es utilizada para poder recomendar películas. En la siguiente tabla se pueden observar los campos que se van a almacenar de las preferencias del usuario, sus tipos y sus descripciones:

Nombre	Tipo		Descripción
phone	INT	UNSIGNED	Teléfono del usuario
	PrimaryKey		
favourite_directors	BLOB		Lista de directores favoritos
favourite_actors	BLOB		Lista de actores favoritos
favourite_genres	BLOB		Lista de géneros de película favoritos
disliked_directors	BLOB		Lista negra de directores
disliked_actors	BLOB		Lista negra de actores
disliked_genres	BLOB		Lista negra de géneros de películas
Cinemas	BLOB		Lista de cines favoritos

Tabla 0-8 - Preferencias del usuario

FilmCacheStorage

El FilmCacheStorage sirve para almacenar la información de forma temporal de una película. Esto se hace ya que al mostrar las recomendaciones que ver, es necesario pedir todas las posibles películas en los cines elegidos por el usuario y en las horas elegidas.

Esta caché es guardada solo por 24 horas, pero este tiempo de expiración es configurable.

En la siguiente tabla se pueden observar los campos que se van a almacenar de la información temporal de una película, sus tipos y sus descripciones:

Nombre	Tipo		Descripción
film_id	INT	UNSIGNED	Identificador de película
	PrimaryKey		
content	BLOB		JSON con el contenido de la película
expiration	INT UNSIGNED		Fecha de expiración de la caché

Tabla 0-9 - Información temporal de una película

CurrentSession

La clase *currentSession* es una clase realizada mediante el patrón de diseño “*Singleton*” que almacena en todo momento el estado de la sesión actual.

Con ella somos capaces de saber que usuario es el que está llamando y si la llamada está identificada o no.

SessionsBackend

SessionsBackend es un sistema interno que agrupa toda la lógica necesaria para gestionar sesiones, llamadores y sesiones en curso.

Es usado para inicializar la sesión en cada petición, así como para comprobar si está identificada, etc.,

UserBackend

UserBackend es un sistema interno que agrupa toda la lógica necesaria para gestionar la gestión de usuario, así como sus preferencias.

3.4.3 Modelo de datos mixto

El modelo de datos mixto es aquel que requiere tanto información local como información externa.

Este es el caso del *FilmRecommendationCalculator*, el cual tiene la responsabilidad de dada una película de *FilmAffinity* y las preferencias del usuario, puntuar la película y añadirle un nuevo campo, la puntuación por gustos del usuario.

Para crear esta puntuación se hace el siguiente algoritmo.

Se toma como máxima puntuación el número de votos con un mínimo de 50 puntos y máximo de 300 puntos. La puntuación de *FilmAffinity* está en base 10, esta es convertida en base máxima puntuación.

Por cada director favorito que coincida con el director de la película se sumará 20 puntos, con un máximo de 20 puntos de recompensa por directores.

Por cada actor favorito que coincida con los actores de la película se sumará 10 puntos, con un máximo de 20 puntos de recompensa por actores.

Por cada género favorito que coincida con los géneros de la película se sumará 10 puntos, con un máximo de 20 puntos de recompensa por género.

Por cada director de la lista negra que coincida con el director de la película se restará 20 puntos, con un máximo de 20 puntos de recompensa por directores.

Por cada actor de la lista negra que coincida con los actores de la película se restará 10 puntos, con un máximo de 20 puntos de recompensa por actores.

Por cada género de la lista negra que coincida con los géneros de la película se restará 10 puntos, con un máximo de 20 puntos de recompensa por género. Esta puntuación final se vuelve a poner en base 10.

Ejemplo:

- Película Votación en FilmAffinity: 6
- Total de votos en FilmAffinity 200.
- Directores favoritos coincidentes = 1
- Actores favoritos coincidentes = 3
- Géneros favoritos coincidentes = 0
- Directores lista negra coincidentes = 0
- Actores lista negra coincidentes = 1
- Géneros lista negra coincidentes = 1

Cálculo del Algoritmo:

Votación FilmAffinity $(6 * 200 / 10)$ + Directores favoritos (20) + actores favoritos (20) – actores lista negra (10) – genero lista negra (10) = 140.

La puntuación final se vuelve a poner en base 10:

$$140 * 10 / 200 = 7$$

La puntuación final sería: 7.

3.5 Diagrama de secuencia del sistema

En el siguiente diagrama de secuencia se puede observar la interacción de las distintas partes de la infraestructura digital al recibir una llamada del usuario:

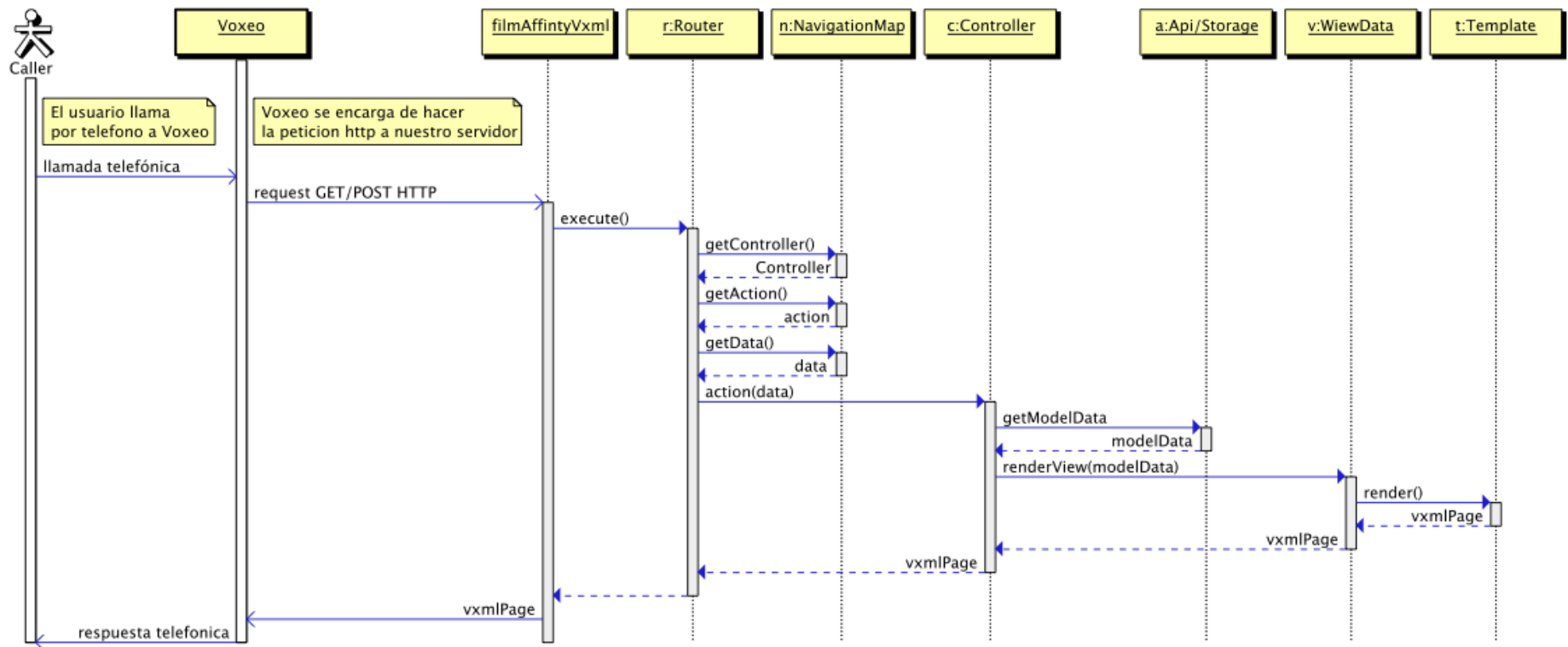


Figura 0-5 - Diagrama de secuencia del sistema

CAPÍTULO 4. Descripción detallada del sistema

A continuación se mostrarán todos los flujos de la aplicación así como ejemplos varios.

Todas las páginas tienen la opción de volver al menú principal pulsando la tecla cero (0) y volver a la página anterior, si existe, pulsando el asterisco (*), para así simular la navegación de un móvil.

4.1 Menú Principal

En el menú principal se muestra las opciones principales para usuarios identificados y anónimos. Estas opciones principales darán lugar a otros menús descritos en los siguientes puntos.

A continuación, se muestra el diagrama de flujo del Menú principal:

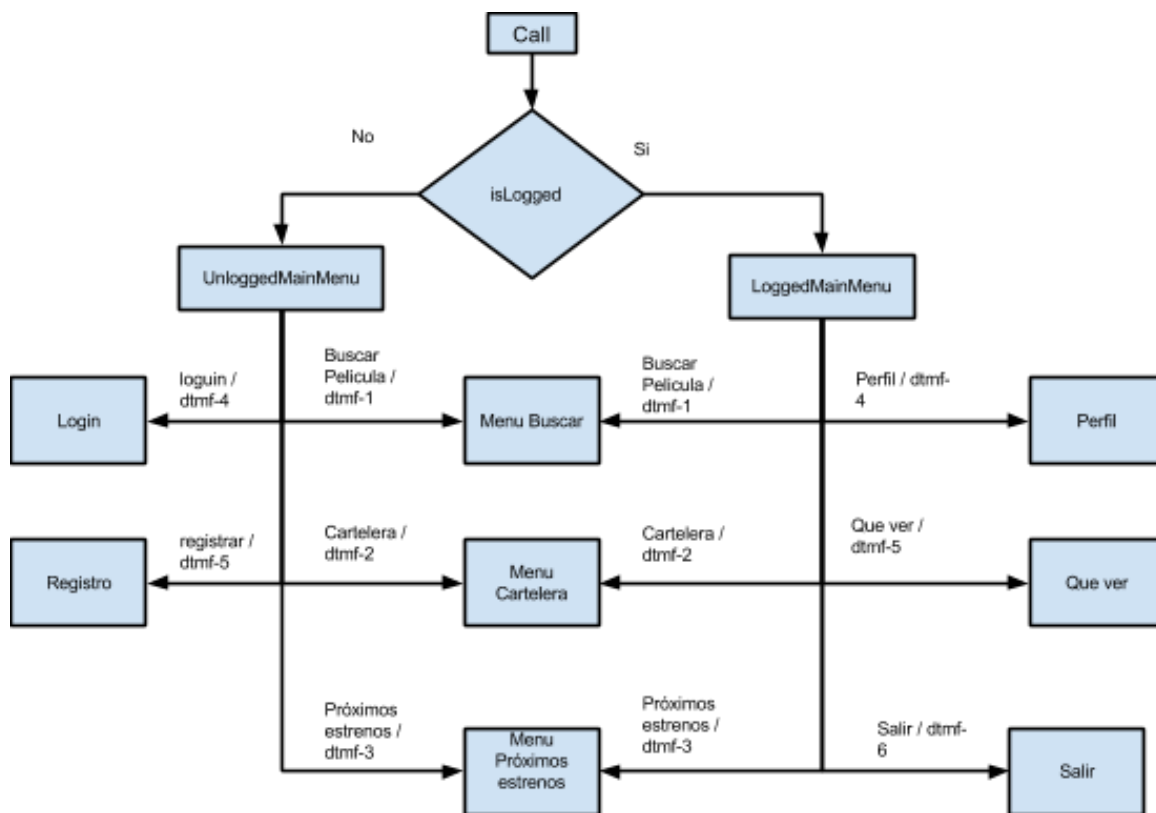


Figura 0-1 - Diagrama de flujo del Menú principal

4.1.1 Diálogos

Los diálogos en el menú principal son básicamente dos: uno para mostrar el menú para usuarios identificados y otro para mostrar el menú para usuarios anónimos.

A continuación, se puede ver el diálogo del menú para usuarios identificados:

Bienvenido al sistema de información de películas por teléfono. Para buscar una película pulse 1 o diga buscar películas. Para ir a la cartelera pulse 2 o diga cartelera. Para ir a los Próximos estrenos pulse 3 o diga Próximos estrenos. Para hacer identificación pulse 4 o diga identificación. Para registrarse pulse 5 o diga registrar.

Tabla 0-1 - Diálogo Menú principal para usuarios anónimos

En la siguiente tabla se puede observar el diálogo del menú principal para usuarios identificados:

Bienvenido al sistema de información de películas por teléfono. Para buscar una película pulse 1 o diga buscar películas. Para ir a la cartelera pulse 2 o diga cartelera. Para ir a los próximos estrenos pulse 3 o diga Próximos estrenos. Para oír que le recomendamos ver pulse 4 o diga que ver. Para ver y modificar su perfil pulse 5 o diga perfil. Para salir de su cuenta pulse 6 o diga salir

Tabla 0-2 - Diálogo Menú principal para usuarios identificados

4.2 Menú Buscar

En el menú buscar se puede elegir el tipo de búsqueda. Buscar por actor, director o título.

Este menú es accedido desde el menú principal.

Todas las búsquedas dirigirán al usuario hacia un Listado paginado de películas el cual se detallará más adelante.

A continuación, se muestra el diagrama de flujo del menú buscar:

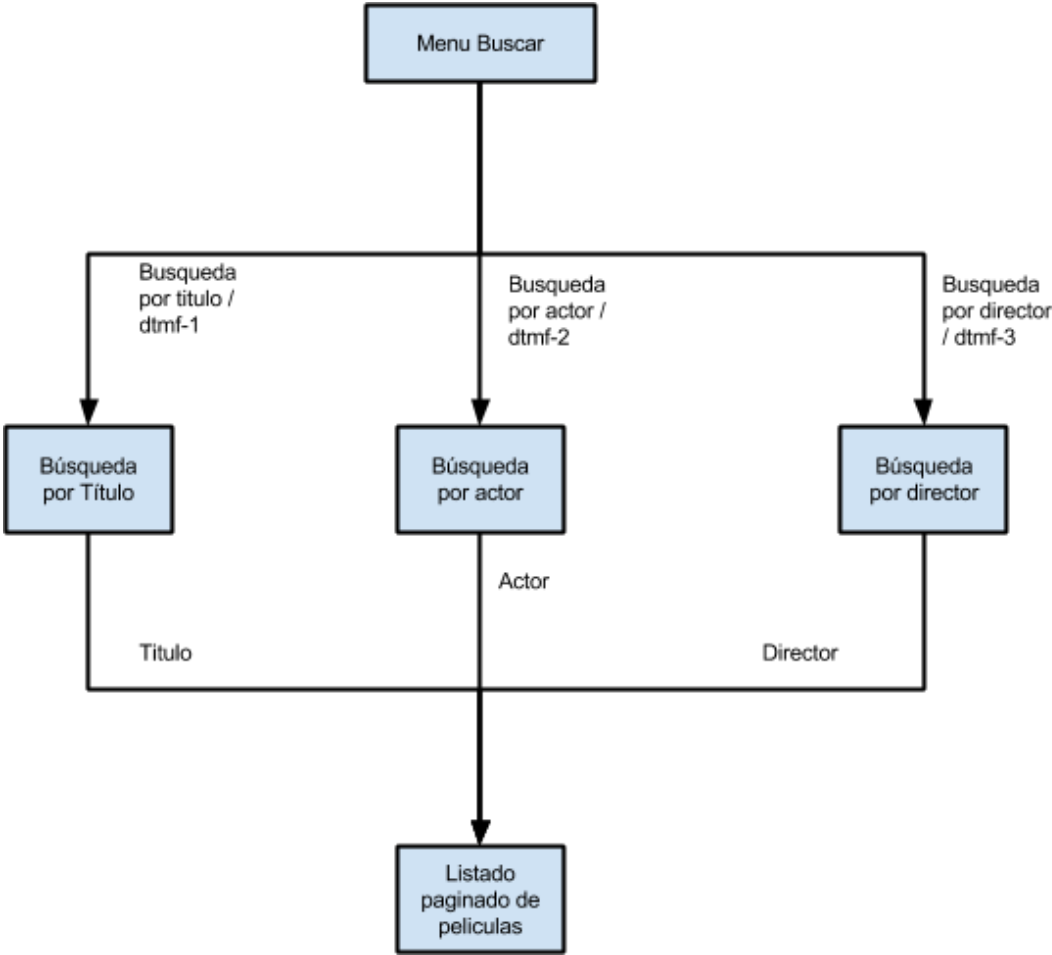


Figura 0-2 - Diagrama de flujo Menú Buscar

Diálogos

Los diálogos en el menú buscar orienta al usuario para elegir el tipo de búsqueda que desea realizar y le ayuda a realizar la búsqueda.

Los diálogos están divididos en el propio menú buscar donde se presentan las opciones de búsqueda que tiene el usuario, y las propias búsquedas.

A continuación, se puede ver el diálogo correspondiente al menú buscar:

Búsqueda de películas: Para buscar una película por título pulse 1 o diga búsqueda por título. Para buscar una película por actor pulse 2 o diga búsqueda por actor. Para buscar una película por director pulse 3 o diga búsqueda por director.

Tabla 0-3 - Diálogo Menú Buscar

Las búsquedas tanto por título, por actor, como por director tienen diálogos muy parecidos en los que se anuncia el tipo de búsqueda que se va a proceder a hacer y se pide el término a buscar.

A continuación, mostramos los tres tipos de búsquedas.

Búsqueda por título. Por favor diga el título a buscar

Tabla 0-4 - Diálogo Búsqueda por título

Búsqueda por actor. Por favor diga el actor a buscar

Tabla 0-5 - Diálogo Búsqueda por actor

Búsqueda por director. Por favor diga el director a buscar

Tabla 0-6 - Diálogo Búsqueda por director

4.3 Menú Cartelera

En el menú cartelera el usuario podrá acceder a las películas actualmente en cartelera ordenadas de varias maneras:

- Ordenadas por puntuación
- Ordenadas por número de votos (popularidad)
- Ordenadas por fecha de estreno

Todas las ordenaciones de la cartelera mostrará el listado paginado de películas en cartelera en el orden mostrado.

A continuación, se muestra el diagrama de flujo del Menú cartelera:

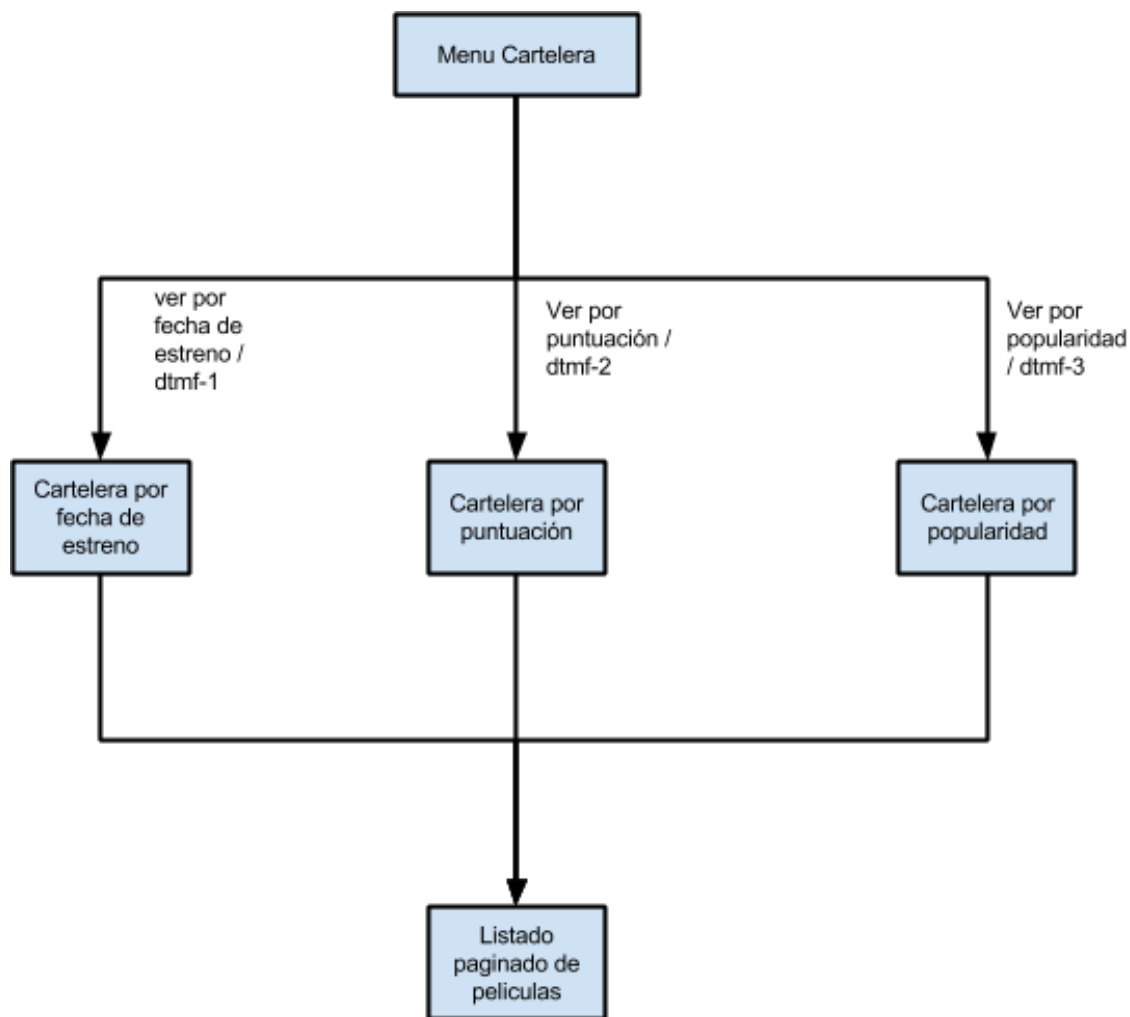


Figura 0-2 - Diagrama de flujo Menú cartelera

4.3.1 Diálogos

Los diálogos del menú cartelera informará al usuario de las distintas opciones de ordenación de las películas elegidas.

A continuación, se puede ver el diálogo correspondiente al menú cartelera:

Menú cartelera: Para ver la cartelera ordenada por fecha de estreno diga ver por fecha de estreno o pulse 1. Para ver la cartelera ordenada por puntuación diga ver por puntuación o pulse 2. Para ver la cartelera ordenada por popularidad (número de votos) diga ver por popularidad o pulse 3.

Tabla 0-7 - Diálogo Menú Cartelera

Una vez elegido el tipo de ordenación en el que se quiere mostrar la información de cartelera, se dirá un título mostrando el tipo de ordenación elegida antes de mostrar el listado de las películas en cartelera.

En las siguientes tablas se pueden ver los títulos que se mostrarán en cada caso:

Películas en cartelera ordenadas por fecha.

Tabla 0-8 - Diálogo Cartelera por fecha de estreno

Películas en cartelera ordenadas por puntuación.

Tabla 0-9 - Diálogo Cartelera por puntuación

Películas en cartelera ordenadas por popularidad.

Tabla 0-10 - Diálogo Cartelera por popularidad

4.4 Menú Próximos Estrenos

En el menú próximos estrenos se podrán consultar los próximos estrenos ordenados de varias maneras:

- Ordenado por puntuación
- Ordenado por número de votos(popularidad)
- Ordenado por fecha de estreno

Todas las ordenaciones de los próximos estrenos mostrará el listado paginado de películas que se estrenarán próximamente en el orden mostrado.

A continuación, se muestra el diagrama de flujo del Menú próximos estrenos:

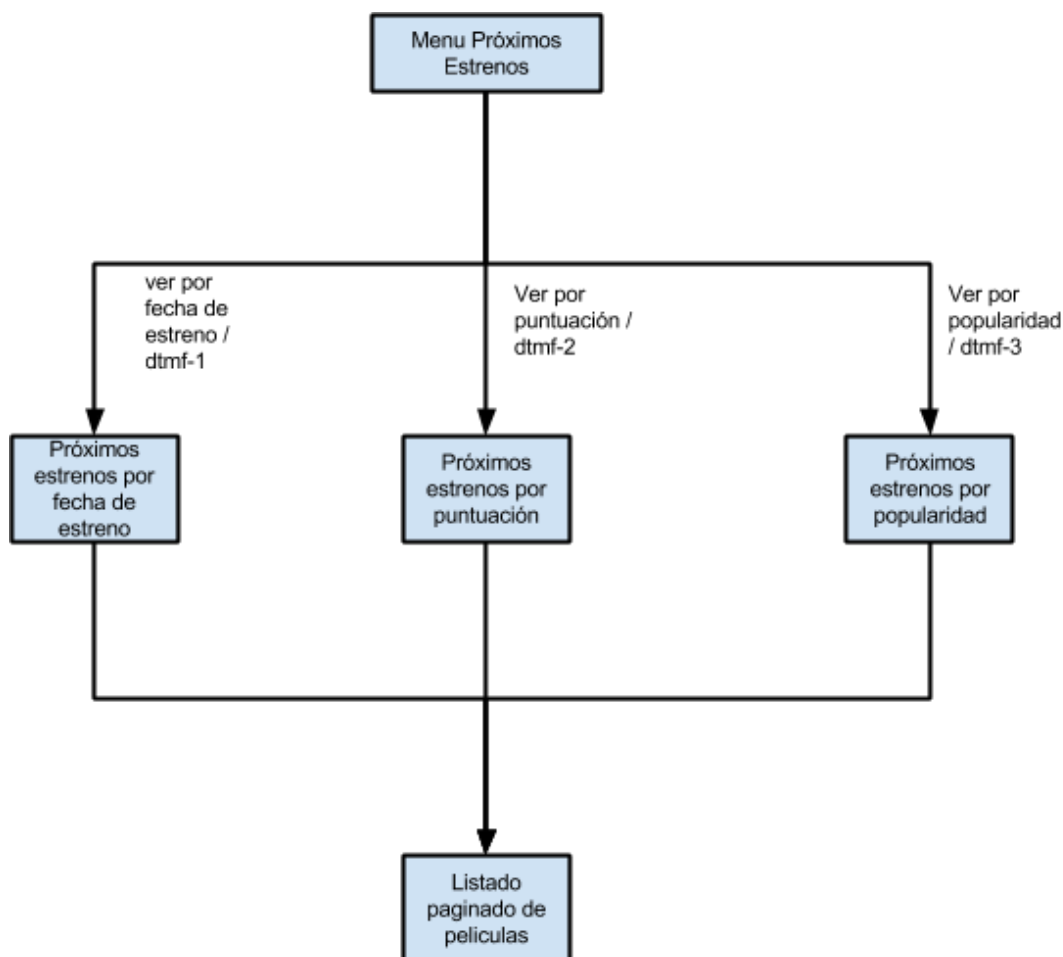


Figura 0-3 - Diagrama de flujo Menú Próximos Estrenos

4.4.1 Diálogos

Los diálogos del menú próximos estrenos informará al usuario de las distintas opciones de ordenación de las películas elegidas.

A continuación, se puede ver el diálogo correspondiente al menú próximos estrenos:

Menú próximos estrenos: Para ver los próximos estrenos ordenados por fecha de estreno diga ver por fecha de estreno o pulse 1. Para ver los próximos estrenos ordenados por puntuación diga ver por puntuación o pulse 2. Para ver los próximos estrenos ordenados por popularidad (número de votos) diga ver por popularidad o pulse 3.

Tabla 0-11 - Diálogo Menú Próximos estrenos

Una vez elegido el tipo de ordenación en el que se quiere mostrar la información de próximos estrenos se dirá un título mostrando el tipo de ordenación elegida antes de mostrar el listado de las películas que se estrenarán próximamente.

En las siguientes tablas se pueden ver los títulos que se mostrarán en cada caso:

Próximos estrenos ordenados por fecha.

Tabla 0-12 - Diálogo Próximos estrenos por fecha de estreno

Próximos estrenos ordenados por puntuación.

Tabla 0-13 - Diálogo Próximos estrenos por puntuación

Próximos estrenos ordenados por popularidad.

Tabla 0-14 - Diálogo Próximos estrenos por popularidad

4.5 Identificación

En la identificación se le dará la opción al usuario de identificarse ya sea por número de teléfono y contraseña, por terminal conocido y contraseña o solo por terminal conocido si la contraseña ha sido guardada con anterioridad.

Una vez que el usuario ha sido identificado correctamente este es redirigido al menú principal.

A continuación, se puede observar el diagrama de flujo de identificación:

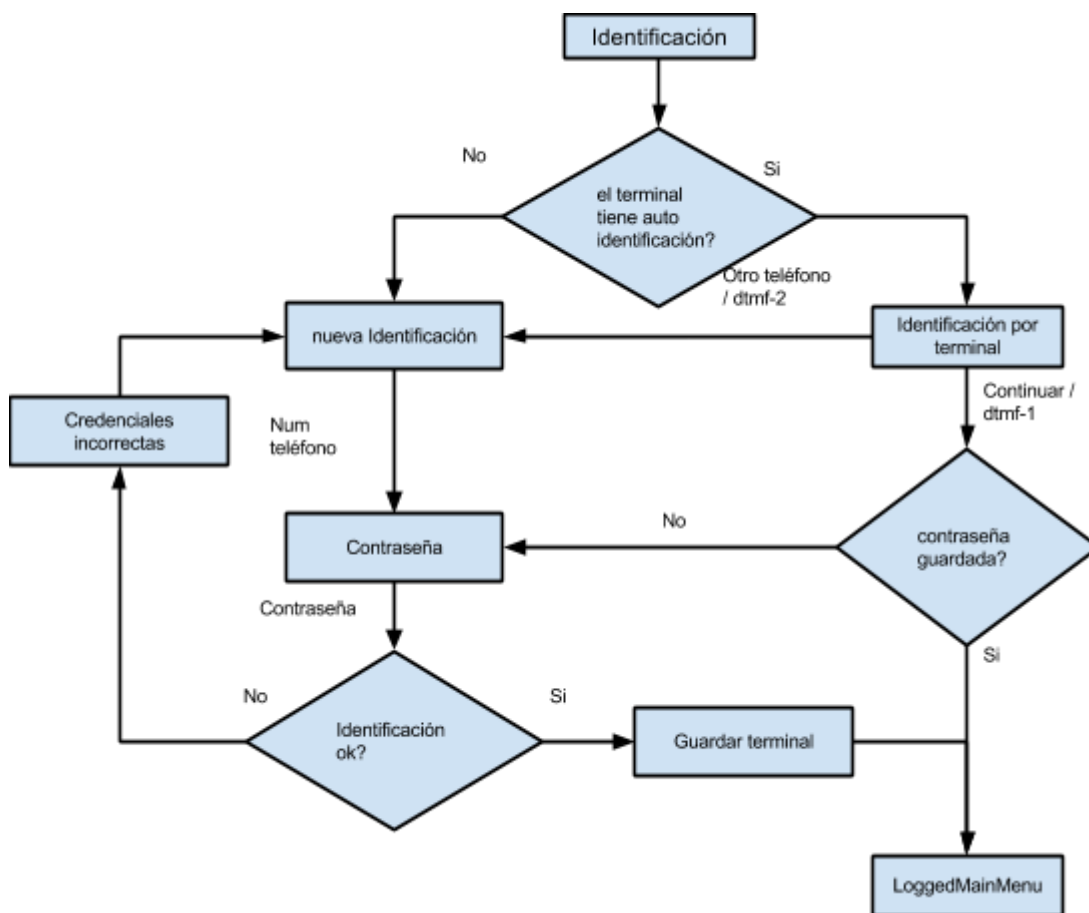


Figura 0-4 - Diagrama de flujo de identificación

4.5.1 Diálogos

Los diálogos del Identificación guiarán al usuario para mostrarle la forma de identificación que quiere efectuar.

Ya que el identificación puede ser hecho por número de teléfono y contraseña o por dispositivo conocido en el cual no habría que introducir el número de teléfono ni el contraseña en el caso de que este último se haya guardado con el dispositivo.

A continuación, se muestran los diálogos para iniciar la identificación del usuario:

Por favor introduzca los 9 dígitos del número de teléfono con el que está registrado

Tabla 0-15 - Diálogo New Identificación

Este terminal está asociado al teléfono: 654321123
Para hacer identificación con este teléfono diga continuar o pulse 1 para usar otro
teléfono diga otro teléfono o marque 2.

Tabla 0-16 - Diálogo Identificación por terminal

Una vez que el usuario ha iniciado el proceso de identificación, se le pedirá la contraseña
con el siguiente diálogo:

Por favor introduzca los 8 dígitos de su contraseña

Tabla 0-17 - Diálogo Contraseña

Si la contraseña y el número de teléfono no concuerdan, se le informará al usuario
mediante el siguiente diálogo de error:

Contraseña o teléfono equivocado.

Tabla 0-18 - Diálogo Identificación incorrecto

Una vez que la identificación ha sido completada satisfactoriamente, se le mostrará al
usuario la opción de guardar la información de identificación para el dispositivo en el que
el usuario se está identificando.

*¿Quieres guardar tu información de acceso para este terminal?. Para guardar tu número
y contraseña diga: guardar todo, para guardar solo el número diga: guardar número, si
no quiere guardar nada diga: continuar sin guardar*

Tabla 0-19 - Diálogo Guardar terminal

4.6 Registro

En el Registro se le da la opción al usuario de registrarse. Para ello se le pedirá al usuario el número de teléfono (este debe de no estar ya registrado) un contraseña de 8 dígitos, este contraseña será necesario que sea repetido para comprobar que el usuario no se ha equivocado.

Una vez el usuario se ha registrado el usuario será redirigido al perfil para que complete sus preferencias.

A continuación, se muestra el diagrama de flujo del Registro:

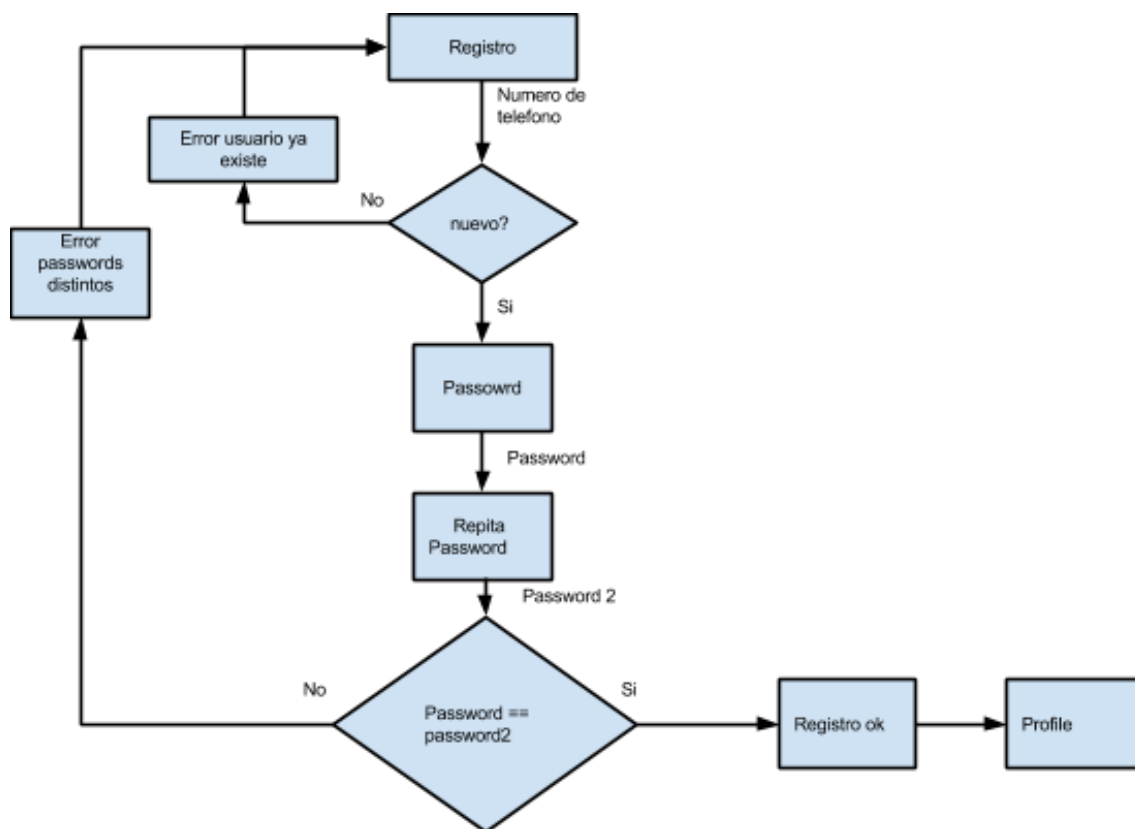


Figura 0-5 - Diagrama de flujo del registro

4.6.1 Diálogos

Los diálogos del registro ayudarán al usuario a completar el registro en la aplicación.

Como primer paso se pide al usuario el número de teléfono que quiere registrar en la aplicación.

Este es el diálogo que se le mostrará al usuario al comenzar el registro.

Por favor introduzca los 9 dígitos del número de teléfono que quiere registrar

Tabla 0-20 - Diálogo Registro

Una vez el usuario ha introducido el número de teléfono que quiere registrar, se le pedirá el contraseña de la cuenta dos veces para comprobar que el usuario no se ha equivocado accidentalmente.

A continuación, se muestra los dos diálogos para pedir al usuario la contraseña de su cuenta:

Por favor introduzca los 8 dígitos de su contraseña

Tabla 0-21 - Diálogo Contraseña

Por favor, vuelva a introducir los 8 dígitos de su contraseña de nuevo

Tabla 0-22 - Diálogo Repita Contraseña

El usuario al introducir el número de teléfono que quiere registrar y las contraseñas puede cometer dos fallos. Que el número de teléfono que se quiere registrar ya esté registrado o que las dos contraseñas introducidas no coincidan. En ambos casos se mostrará un error y se le volverá a pedir la información al usuario. En las siguientes tablas se puede ver los dos mensajes de error mostrados al usuario en cada caso:

Lo sentimos ese teléfono ya se encuentra registrado.

Tabla 0-23 - Diálogo Error Usuario ya existe

Ambas contraseñas deben coincidir.

Tabla 0-24 - Diálogo Error Contraseñas distintos

Una vez que el registro ha sido completado correctamente se le mostrará un mensaje de confirmación al usuario y se le redirigirá al perfil para que el usuario lo rellene.

A continuación, puede ver el mensaje de confirmación de registro:

Usuario creado correctamente, por favor rellene su perfil

Tabla 0-25 - Diálogo Registro Ok

4.7 Perfil

En el perfil el usuario podrá acceder a gestionar sus preferencias. Estas preferencias están divididas en los siguientes subtipos:

- gestionar provincia
- gestionar horario preferido para ir al cine
- gestionar directores
- gestionar actores
- gestionar géneros
- gestionar cines favoritos

Cada uno de los submenús para gestionar cada una de las preferencias serán descritas con detalle más adelante.

En el siguiente diagrama de flujo se puede ver la funcionalidad que ofrece el menú Perfil:

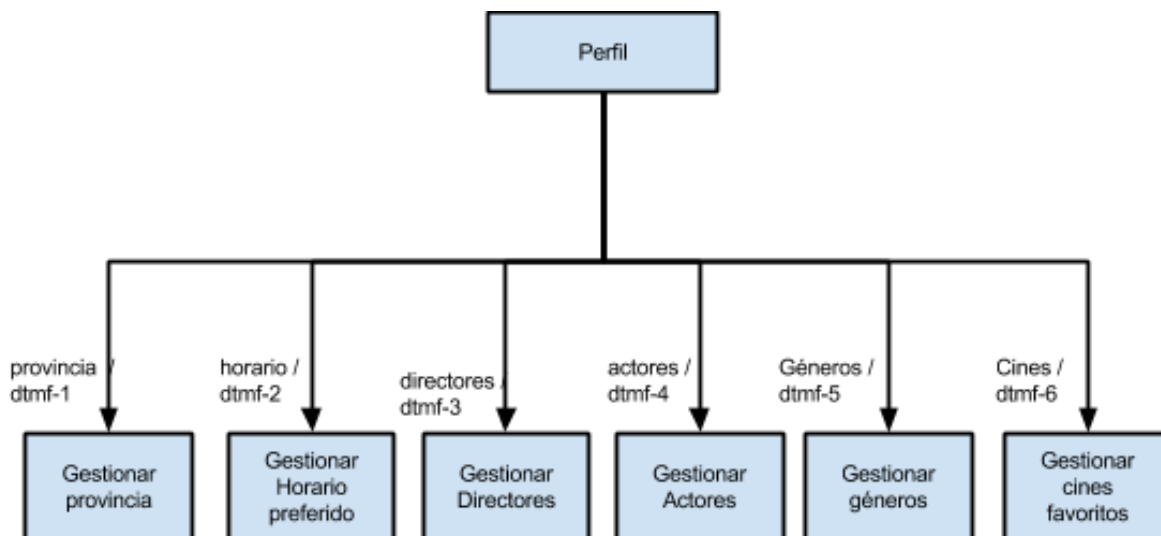


Figura 0-6 - Diagrama de flujo Perfil

4.7.1 Diálogos

En este menú se le mostrará al usuario todas las preferencias que puede gestionar y se le pedirá que elija una para entrar en su gestión.

Se mostrará el siguiente diálogo para mostrarle las opciones validas al usuario:

Bienvenido a su perfil, con él seremos capaces de recomendarle películas Para ver o modificar su provincia, diga provincia o marque 1 Para ver o modificar su horario preferido para ir al cine, diga horario o marque 2 Para ver o modificar sus directores preferidos, diga directores o marque 3 Para ver o modificar sus actores preferidos, diga actores o marque 4 Para ver o modificar sus géneros preferidos, diga géneros o marque 5

Tabla 0-26 - Diálogo Perfil

4.8 Gestionar provincia

En este apartado el usuario será capaz de consultar y modificar su provincia. Esta provincia será usada para ser capaz de recomendarle cines favoritos al usuario.

En la siguiente figura se puede ver el diagrama de flujo de la provincia:

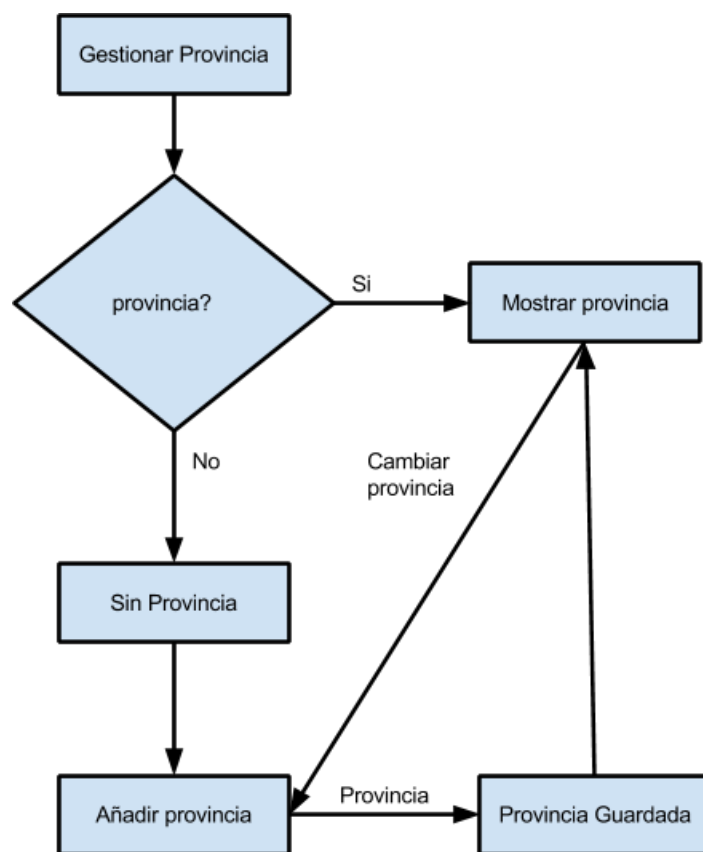


Figura 0-7 - Diagrama de flujo Gestionar Provincia

4.8.1 Diálogos

En el menú gestionar provincia se le mostrará al usuario la actual provincia guardada y se le dará la opción de cambiarla.

En la siguiente tabla se puede ver el diálogo mostrado al usuario para mostrarle su provincia:

<p><i>Tu actual provincia es: Madrid, para cambiarla diga cambiar provincia</i></p>

Tabla 0-27 - Diálogo Mostrar Provincia

En el caso de que el usuario no tenga provincia se le advertirá de ello y se le redirigirá a la página para añadir provincia.

A continuación, se puede ver el diálogo usado para advertir al usuario acerca de que no tiene provincia:

No tienes provincia guardada

Tabla 0-28 - Diálogo Sin Provincia

Una vez en el formulario de añadir provincia, se le pedirá al usuario el nombre de su provincia para añadirla, como se puede ver en el diálogo en la siguiente tabla:

Por favor diga el nombre de su provincia

Tabla 0-29 - Diálogo Añadir Provincia

Una vez que la provincia ha sido guardada se le mostrará un mensaje al usuario y se le redirigirá al inicio del menú.

En la siguiente tabla se puede observar el diálogo usado para mostrar al usuario que la provincia se ha guardado correctamente

Provincia guardada.

Tabla 0-30 - Diálogo Provincia guardada

4.9 Gestionar Horario

En este apartado el usuario podrá consultar y modificar el horario favorito para ir al cine.

Este horario será la hora que empieza su horario favorito y a la hora que acaba este horario, por lo que las películas han de acabar antes del final de su horario favorito.

Con esta información el sistema será capaz de descartar sesiones de cine que no concuerden con los horarios del usuario para recomendarle que ver.

En la siguiente figura se puede observar el diagrama de flujo del menú para gestionar el horario preferido

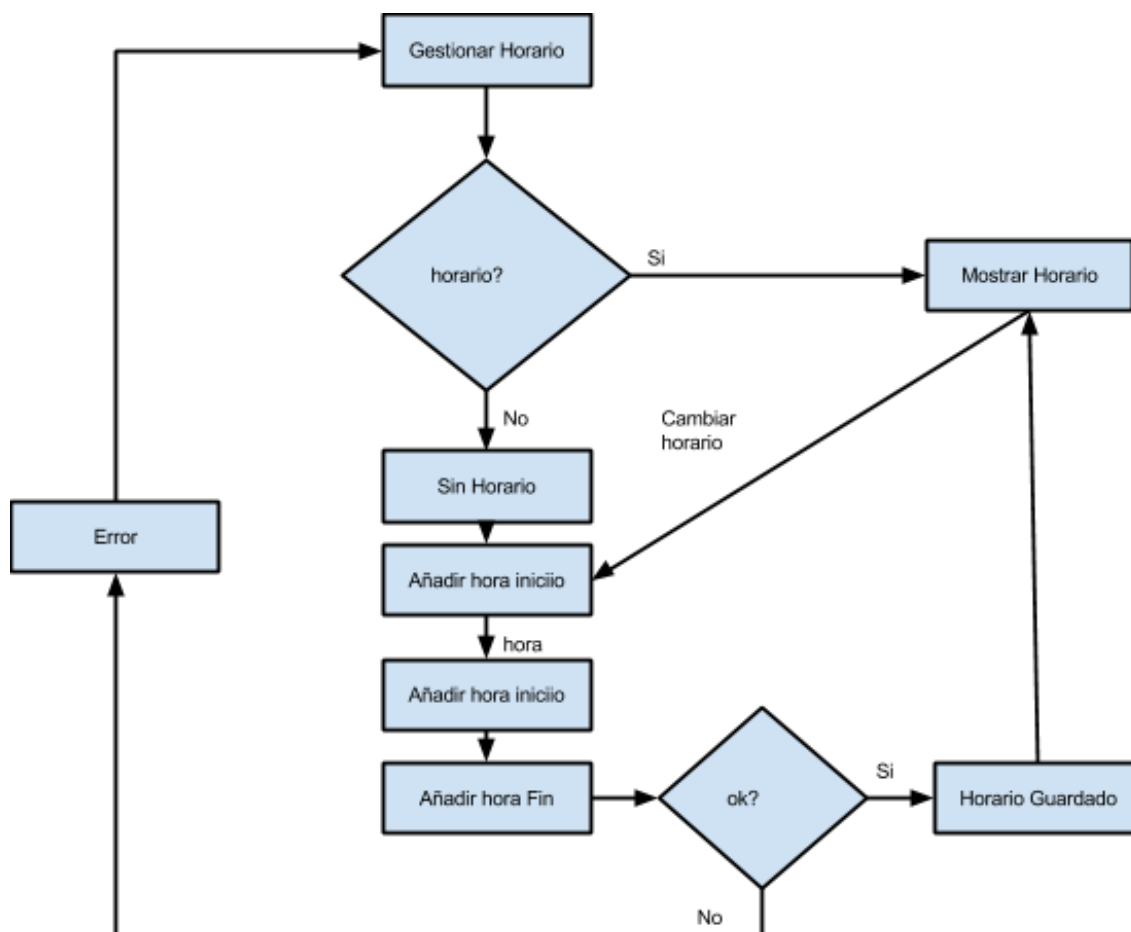


Figura 0-8 - Diagrama de flujo Gestionar horario

4.9.1 Diálogos

Con los diálogos de este menú se le orientará al usuario para introducir las horas favoritas para ir al cine teniendo en cuenta que tiene que ser un horario en el que dé tiempo a ver una película entera (al menos dos horas).

En este menú se le mostrará al usuario el horario que tiene guardado y se le dará la opción de cambiarlo como se puede ver en el diálogo de la siguiente tabla:

Tu horario preferido es de: 19 hasta 1, para cambiarlo diga cambiar horario

Tabla 0-31 - Diálogo Mostrar Horario

Si el usuario no tuviese horario preferido guardado en la aplicación, se le advertirá sobre ello y se le redirigirá a la página para añadir horario con el siguiente diálogo:

No tienes horario preferido guardado.

Tabla 0-32 - Diálogo Sin Horario

Una vez que el usuario accede a la página para añadir el horario preferido para ir al cine, se le pedirá introducir la hora de inicio para ir al cine y tras esto la hora de fin.

En las dos siguientes tablas se pueden observar los diálogos de los formularios para añadir los horarios:

Por favor marque la hora de 0 a 23 a la que empieza su horario preferido para ir al cine

Tabla 0-33 - Diálogo Añadir Hora inicio

Por favor marque la hora de 0 a 23 a la que termina su horario preferido para ir al cine

Tabla 0-34 - Diálogo Añadir Hora Fin

En el caso de que el horario sea inferior a la duración media de una película (dos horas), se mostrará al usuario un mensaje de error y se le redirigirá al inicio de la gestión de horario preferido.

El mensaje error utilizado se puede ver en la siguiente tabla:

El horario preferido debe de tener un rango de más de dos horas

Tabla 0-35 - Diálogo Error

Una vez que el horario ha sido guardado correctamente, se mostrara el siguiente mensaje de confirmación y se redirigirá al usuario al inicio de la gestión de horario preferido:

Horario guardado

Tabla 0-36 - Diálogo Horario guardado

4.10 Gestionar Directores

En este apartado el usuario podrá consultar, borrar y añadir directores a su lista de directores favoritos y su lista negra de directores. Esta información será utilizada para poder recomendar películas al usuario.

En la siguiente figura se puede observar el diagrama de flujo de la gestión de directores.

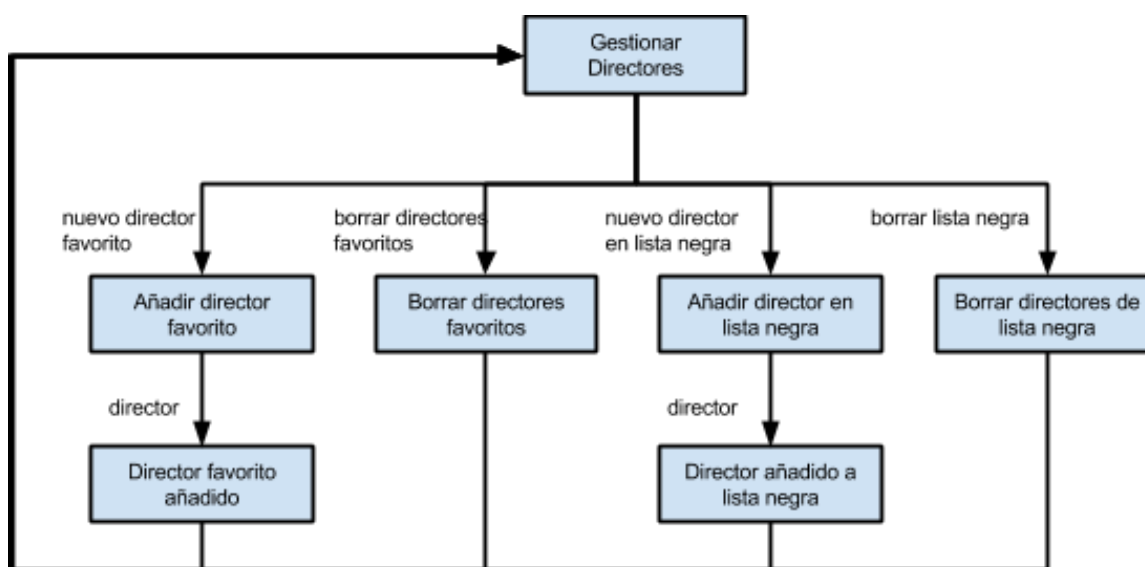


Figura 0-9 - Diagrama de flujo Gestionar Directores

4.10.1 Diálogos

Los diálogos de este menú orientaran al usuario para añadir o borra directores de sus listas de directores (lista favorita y lista negra). Además le mostrara la información sobre los directores que tiene en sus listas

A continuación, se muestran los diálogos del menú gestionar directores en el caso de tener directores guardados y de no tenerlo:

Aún no tienes ningún director favorito guardado, Para añadir un nuevo director diga nuevo director favorito. Aun no tienes lista negra de directores, Para añadir un nuevo director a la lista negra diga nuevo director en lista negra.

Tabla 0-37 - Diálogo Gestionar Directores (Sin directores guardados)

Estos son tus directores favoritos: Alex de la Iglesia, Alejandro Amenábar. Para añadir un nuevo director favorito diga nuevo director favorito, para borrarlos diga borrar directores favoritos

Esta es su lista negra de directores: Pedro Almodóvar, Julio Salinas. Para añadir un nuevo director a la lista negra diga nuevo director en lista negra, para borrarlos diga borrar lista negra

Tabla 0-38 - Diálogo Gestionar Directores (Con directores guardados)

Si el usuario elige añadir un director, ya sea como favorito o a la lista negra, se mostrará el mismo diálogo como se muestra a continuación:

Por favor diga el nombre del director

Tabla 0-39 - Diálogo Añadir director (favorito o lista negra)

Una vez que el director ha sido añadido, se le mostrará un diálogo de confirmación y se le redirigirá al inicio de la gestión de directores.

Dependiendo si el director ha sido añadido a los directores favoritos o a la lista negra se mostrará uno de los siguientes diálogos:

Director añadido a la lista de directores favoritos.

Tabla 0-40 - Diálogo Director favorito añadido

Director añadido a la lista negra de directores.

Tabla 0-41 - Diálogo Director añadido a lista negra

Si el usuario elige borrar los directores, se mostraran los siguientes mensajes de confirmación de borrado y se les redirigirá hacia el inicio de la gestión de directores:

Lista de directores favoritos borrada

Tabla 0-42 - Diálogo Borrar Directores favoritos

Lista negra de directores borrada

Tabla 0-43 - Diálogo Lista negra de directores borrada

4.11 Gestionar Actores

En este apartado el usuario podrá consultar, borrar y añadir actores a su lista de directores favoritos y su lista negra de actores.

Esta información será utilizada para poder recomendar películas al usuario.

En la siguiente figura se puede observar el diagrama de flujo de la gestión de actores

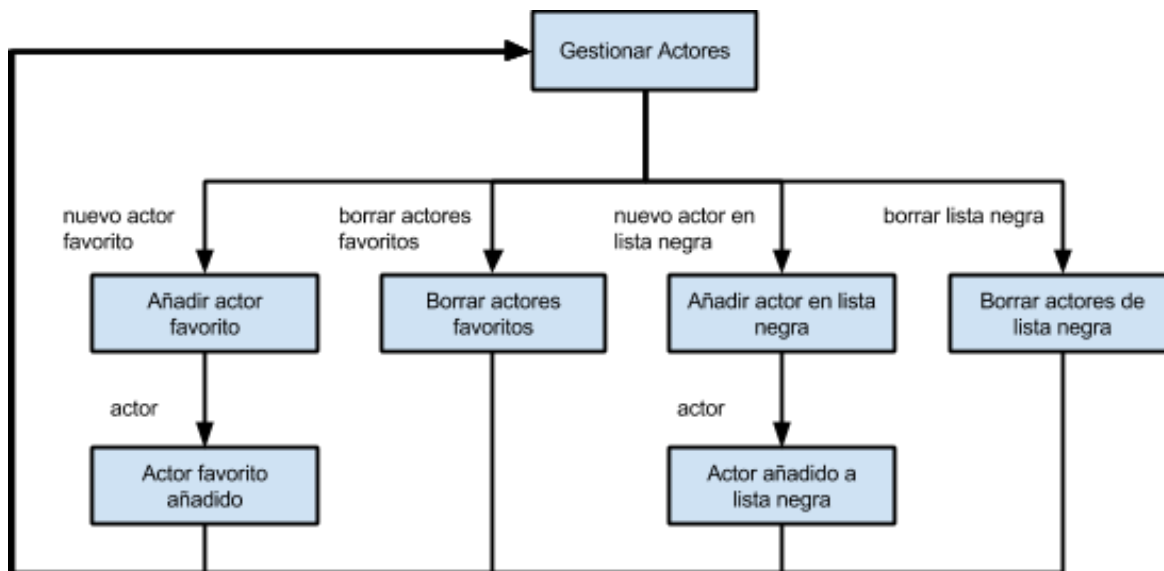


Figura 0-10 - Diagrama de flujo Gestionar Actores

4.11.1 Diálogos

Los diálogos de este menú orientaran al usuario para añadir o borra actores de sus listas de actores (lista favorita y lista negra). Además le mostrara la información sobre los actores que tiene en sus listas

A continuación, se muestran los diálogos del menú gestionar actores en el caso de tener actores guardados y de no tenerlo:

Aún no tienes ningún actor favorito guardado, Para añadir un nuevo actor diga nuevo actor favorito. Aun no tienes lista negra de actores, Para añadir un nuevo actor a la lista negra diga nuevo actor en lista negra.

Tabla 0-44- Diálogo Gestionar Actores (Sin actores guardados)

Estos son tus actores favoritos: Alex de la Iglesia, Alejandro Amenábar. Para añadir un nuevo actor favorito diga nuevo actor favorito, para borrarlos diga borrar actores favoritos

Esta es su lista negra de actores: Pedro Almodóvar, Julio Salinas. Para añadir un nuevo actor a la lista negra diga nuevo actor en lista negra, para borrarlos diga borrar lista negra

Tabla 0-45 - Diálogo Gestionar Actores (Con actores guardados)

Si el usuario elige añadir un actor, ya sea como favorito o a la lista negra, se mostrará el mismo diálogo como se muestra a continuación

Por favor diga el nombre del actor

Tabla 0-46 - Diálogo Añadir actor (favorito o lista negra)

Una vez que el actor ha sido añadido, se le mostrará un diálogo de confirmación y se le redirigirá al inicio de la gestión de actores.

Dependiendo si el actor ha sido añadido a los actores favoritos o a la lista negra se mostrará uno de los siguientes diálogos:

Actor añadido a la lista de actores favoritos.

Tabla 0-47 - Diálogo Actor favorito añadido

Actor añadido a la lista negra de actores.

Tabla 0-48 - Diálogo Actor añadido a lista negra

Si el usuario elige borrar los actores, se mostraran los siguientes mensajes de confirmación de borrado y se les redirigirá hacia el inicio de la gestión de actores:

Lista de actores favoritos borrada

Tabla 0-49 - Diálogo Borrar Actores favoritos

Lista negra de actores borrada

Tabla 0-50 - Diálogo Lista negra de actores borrada

4.12 Gestionar Cines

En este apartado el usuario podrá consultar modificar, borrar y añadir cines a su lista de cines favoritos. Los cines favoritos serán usados para recomendar películas que se proyecten en esos cines. En la siguiente figura se puede ver el diagrama de flujo del menú para gestionar los cines favoritos:

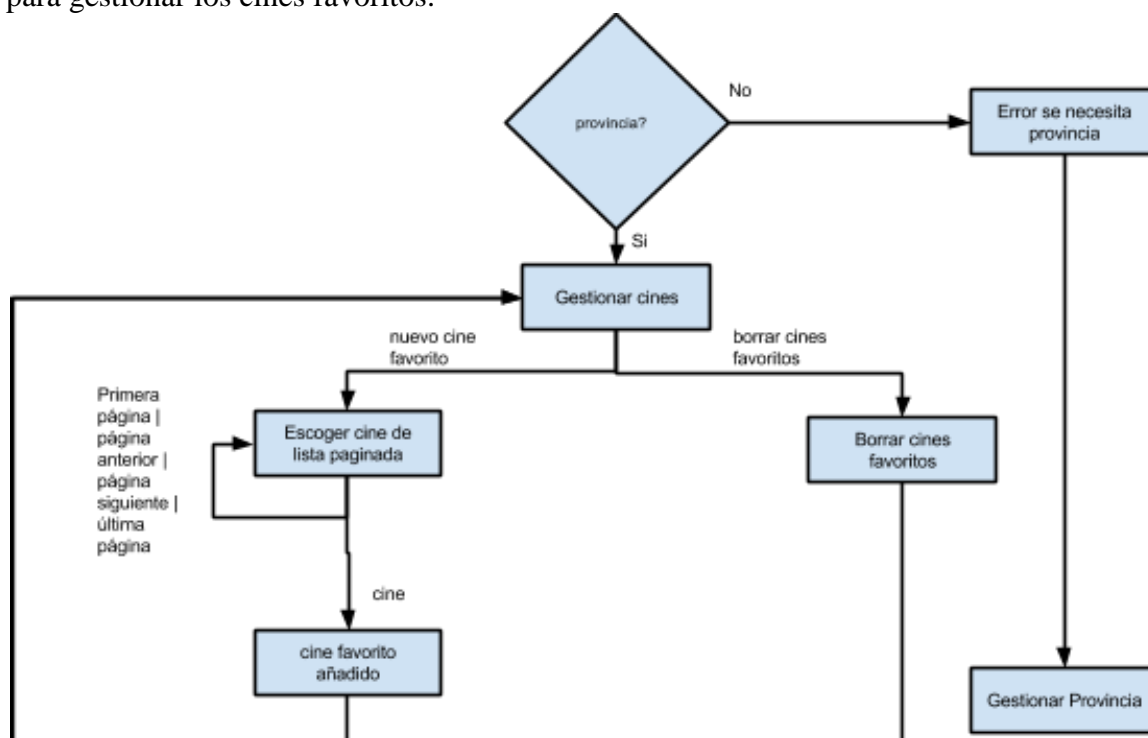


Figura 0-11 - Diagrama de flujo Gestionar Cines

4.12.1 Diálogos

El menú gestionar cines ayudará al usuario a consultar sus cines favoritos y a gestionarlos.

El usuario podrá añadir cines de la provincia que tiene guardada y borrar todos los cines guardados.

Al inicio del menú mostrará al usuario los cines que tiene guardados y las opciones que tiene el usuario como se puede observar en las siguientes tablas:

Aún no tienes ningún cine favorito guardado. Para añadir un nuevo cine di nuevo cine

Tabla 0-51 - Diálogo Gestionar Cines (sin cines guardados)

Esta es la lista de tus cines favoritos: Cinema parquesur. Para añadir un nuevo cine di nuevo cine, para borrarlos di borrar cines

Tabla 0-52 - Diálogo Gestionar Cines (Con cines guardados)

Si el usuario decide añadir un cine, se le mostraran los cines de su provincia de forma paginada de 10 en 10 de la manera que se muestra en el siguiente menú:

Mostrando página 0 de 5. La Dehesa Cuadernillos, para elegirlo diga o marque 1. Cinesa La Moraleja, para elegirlo diga o marque 2. Yelmo Cines Tresaguas, para elegirlo diga o marque 3. Cinesa Xanadu, para elegirlo diga o marque 4. Yelmo Cines Planetocio, para elegirlo diga o marque 5. Cinesa Loranca, para elegirlo diga o marque 6. Cinesa Nassica-Getafe, para elegirlo diga o marque 7. Cinesa Las Rozas - Heron City, para elegirlo diga o marque 8. Cinesa Parquesur, para elegirlo diga o marque 9.

Tabla 0-53 - Diálogo Escoger cine de lista paginada

Una vez que el usuario elige un cine el sistema mostrara el siguiente diálogo de confirmación y se le redirigirá al inicio de la gestión de cines:

Cine añadido a cines favoritos.

Tabla 0-54 - Diálogo Cine favorito añadido

En el caso de que el usuario elija borrar los cines, se le mostrará el siguiente diálogo y se le redirigirá al inicio de la gestión de cines.

Lista de cines favoritos borrada

Tabla 0-55 - Diálogo Borrar cines favoritos

En el caso de que el usuario no tenga una provincia guardada, no se le podrán mostrar cines por lo que se le mostrara un mensaje de error y se le redirigirá a la gestión de provincias:

Necesitamos saber tu provincia para poder buscar cines.

Tabla 0-56 - Diálogo Error se necesita provincia

4.13 Que Ver (Sistema de recomendación de películas)

El sistema de recomendación de películas busca en la lista de tus cines favoritos, las películas que se emiten en los tu horario preferido (si no está definido se buscará en cualquiera). Estas películas se ordenan por recomendación personal si es calculable y si no por puntuación de FilmAffinity.

4.13.1 Sistema de recomendación de películas

El sistema de recomendación de películas calcula la puntuación de la siguiente manera:

La puntuación se calcula por pesos de cada una de las preferencias del usuario de manera que se calcula una puntuación máxima que puede conseguir una película con la puntuación conseguida y se transforma a puntuación sobre 10.

la puntuación de FilmAffinity se convierte de base 10 a una base entre 50 y 300 dependiendo del número de votos, de esta manera para una puntuación con 100 votos tendrá como máximo 100 puntos de peso, si la puntuación es un 6, se convertirá a 60 de puntuación y 100 de puntuación máxima.

Si el director de la película es uno de tus directores favoritos se sumaran 20 puntos a la puntuación.

Si el director de la película está en la lista negra de directores se restaran 20 puntos de la puntuación

Por cada actor de la película que sea uno de tus actores favoritos se sumarán 10 puntos a la puntuación hasta un máximo de 20 puntos.

Por cada actor de la película que esté en tu lista negra de actores se restará 10 puntos a la puntuación hasta un máximo de 20 puntos.

Por cada género de la película que sea uno de tus géneros favoritos se sumarán 10 puntos a la puntuación hasta un máximo de 20 puntos.

Por cada género de la película que esté en tu lista negra de géneros se restará 10 puntos a la puntuación hasta un máximo de 20 puntos.

Si no se ha conseguido restar o sumar nada a la puntuación original de FilmAffinity se supone que no se tiene suficiente información en el perfil como para poder hacer la recomendación. Todos los pesos son configurables.

En la siguiente figura se muestra el diagrama de flujo del menú que ver:

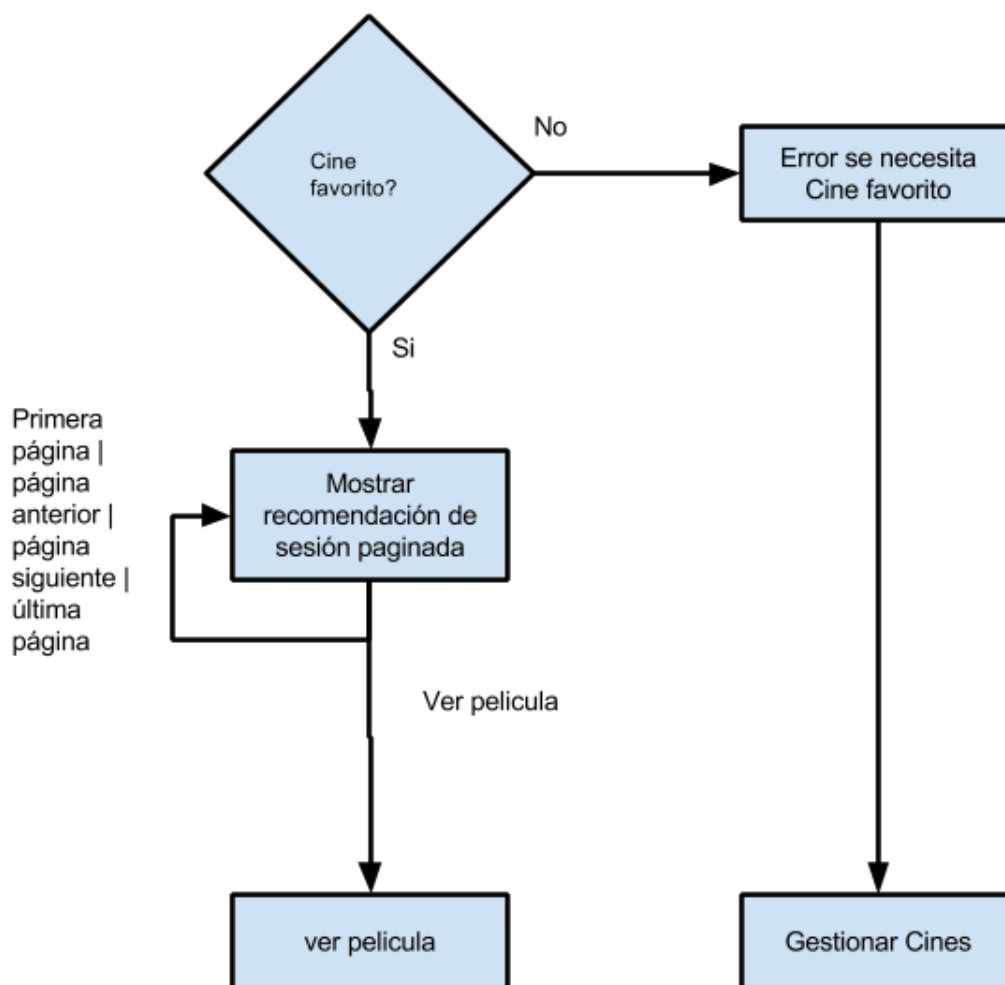


Figura 0-12 - Diagrama de flujo Que ver

4.13.2 Diálogos

En este apartado los diálogos mostrarán al usuario cuales son las películas que debería de ir a ver teniendo en cuenta sus gustos.

Para poder recomendarle proyecciones de cine el usuario debe de tener guardado al menos un cine favorito, por lo que si no tiene uno se le mostrará un mensaje de error con el siguiente diálogo.

Para poder recomendarte sesiones necesitamos saber tus cines favoritos

Tabla 0-57 - Diálogo Error se necesita cines favoritos.

Si el usuario tiene un cine pero aun así no se encuentra ninguna proyección de película se le mostrará un mensaje que le indicará que quizás debería de cambiar su perfil. Como se puede observar en la siguiente tabla.

No hemos encontrado ninguna recomendación por favor modifica tu perfil.

Tabla 0-58 - Diálogo Mostrar recomendación de sesión paginadas (Sin sesiones)

En el caso de que el sistema sí que tenga proyecciones de películas para recomendar estas serán mostradas paginadas de 1 en 1, con una pequeña descripción de la película:

Mostrando película 1 de 17 ordenadas por recomendación.
Título: Misión imposible: Nación secreta
Género: Acción. Thriller, Espionaje. Secuela
Director: Christopher McQuarrie
Puntuación de FilmAffinity: 6,7sobre 10 de 3895 votos
Puntuación sobre tus gustos: 7,13 sobre 10
Proyecciones: Cine Cinesa Xanadú. Sesiones: domingo 30/08/2015 a las 4 y cuarto, 7 y 10, 10 en punto. Lunes 31/08/2015 a las 4 y cuarto, 7 y 10, 10 en punto. Martes 1/09/2015 a las 4 y cuarto, 7 y 10, 10 en punto. Miércoles 2/09/2015 a las 4 y cuarto, 7 y 10, 10 en punto. Jueves 3/09/2015 a las 4 y cuarto, 7 y 10, 10 en punto.
Sinopsis: Con la FMI disuelta y Ethan Hunt (Tom Cruise) abandonado a su suerte, el equipo tiene que enfrentarse contra el Sindicato, una red de agentes especiales altamente preparados y entrenados. Estos grupos están empeñados en crear un nuevo orden mundial mediante una serie de ataques terroristas cada vez más graves. Ethan reúne a su equipo y une sus fuerzas con la agente británica renegada Ilsa Faust (Rebecca Ferguson), quien puede que sea o no miembro de esta nación secreta, mientras el grupo se va enfrentando a su misión más imposible hasta la fecha... Quinta entrega de la saga Misión Imposible. (FILMAFFINITY). Para ver la ficha completa de la película diga ver ficha de película o pulse 1 Ir a la primera Película, para elegirlo diga primera Película. Ir a la Película anterior, para elegirlo diga Película anterior. Ir a la Película siguiente, para elegirlo diga Película siguiente. Ir a la última Película, para elegirlo diga última Película.

Tabla 0-59 - Diálogo Mostrar recomendación de sesión paginadas

4.14 Listado paginado de películas.

El usuario será capaz de escuchar un listado de películas paginadas, es decir serán varias películas en varias páginas.

Este menú le permitirá elegir una película de la lista o moverse por las páginas.

En la siguiente figura se muestra el diagrama de flujo del Listado paginado de películas:

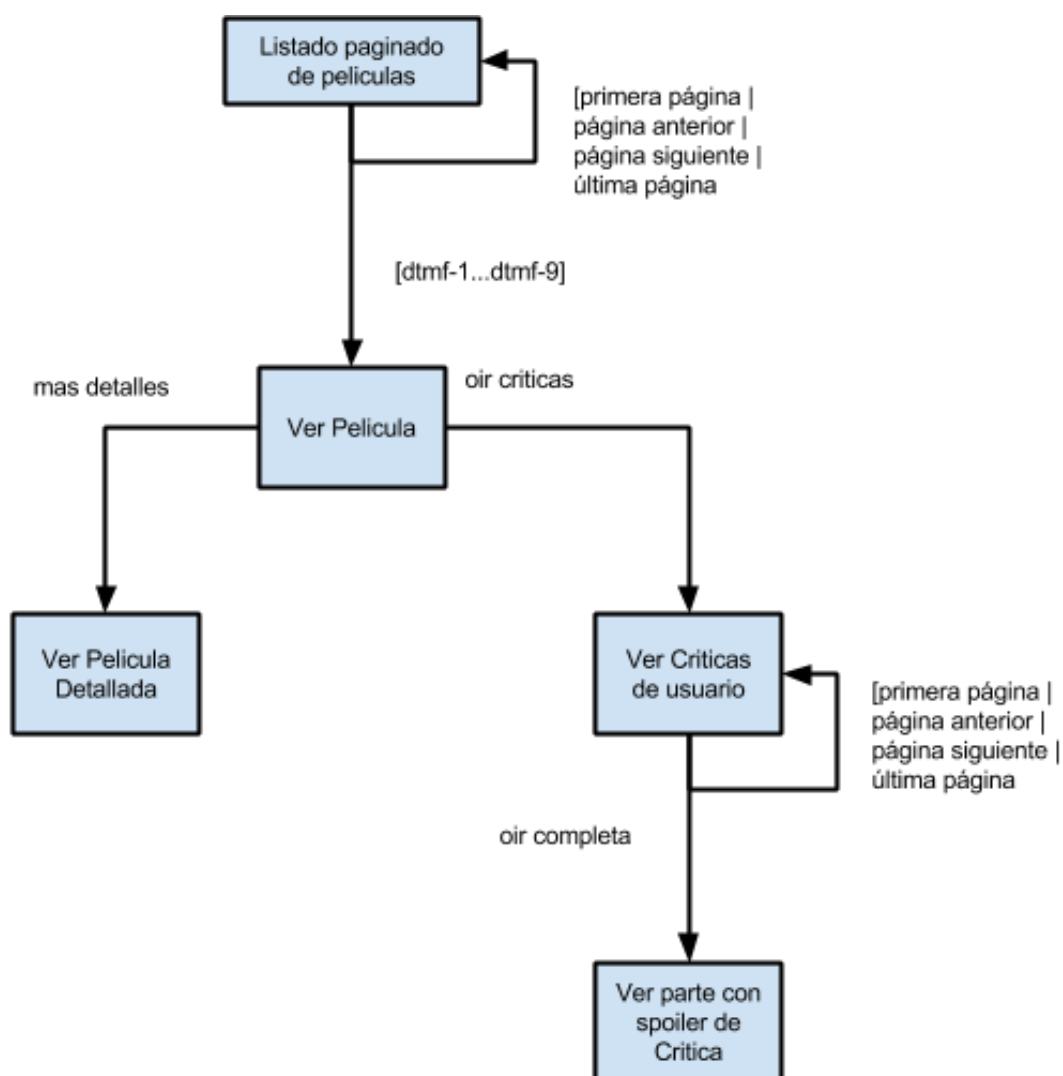


Figura 0-13 - Diagrama de flujo Listado de películas

4.14.1 Diálogos

Los listados de películas serán mostrados de 10 en 10 y serán seleccionables por medio de las teclas del teléfono o diciendo el número deseado.

A continuación, se muestra un diálogo mostrando un listado paginado de películas:

Mostrando página 2 de 4.
Amy La chica detrás del nombre, para elegirlo diga o marque 1.
Ciudades de papel, para elegirlo diga o marque 2.
Y de repente tú, para elegirlo diga o marque 3.
Una historia real, para elegirlo diga o marque 4.
Cuatro Fantásticos, para elegirlo diga o marque 5.
Les combattants, para elegirlo diga o marque 6.
Rey Gitano, para elegirlo diga o marque 7.
La horca, para elegirlo diga o marque 8.
Aprendiendo a conducir, para elegirlo diga o marque 9.
Ir a la primera página, para elegirlo diga primera página.
Ir a la página anterior, para elegirlo diga página anterior.
Ir a la página siguiente, para elegirlo diga página siguiente.
Ir a la última página, para elegirlo diga última página.

Tabla 0-60 - Diálogo Listado paginado de películas

Una vez que el usuario elije una descripción de una película esta puede ser mostrada de varias maneras dependiendo del usuario.

A continuación, mostramos ejemplos de descripciones de películas para un usuario no identificado, un usuario identificado sin preferencias y un usuario identificado con las preferencias configuradas:

Información sobre la película: Cuatro Fantásticos
Título: Cuatro Fantásticos
Género: Ciencia ficción. Fantástico. Acción, Superhéroes. Cómic. Marvel Comics
Director: Josh Trank
Puntuación de FilmAffinity: 3.8 sobre 10 de 695 votos
Puntuación sobre tus gustos: Por favor identificate o regístrate para poder recomendarte películas dependiendo tus gustos.
Fecha de estreno: 21/08/15
Año: 2015
Sinopsis: Cuatro jóvenes inadaptados se teletransportan a un peligroso universo alternativo, lo que les confiere extraños poderes. Cuando sus vidas cambian de forma drástica e irremediable, deben aprender a controlar sus nuevas habilidades y trabajar en equipo para salvar al mundo de un viejo conocido que ahora se ha convertido en un

temible enemigo. Adaptación contemporánea del equipo de superhéroes de Marvel. (FILMAFFINITY)

Esta película contiene 37 críticas de usuarios, para oírlas diga oír críticas. Para más detalles de diga más detalles

Tabla 0-61 - Diálogo Ver Película para usuario no identificado

Información sobre la película: Cuatro Fantásticos

Título: Cuatro Fantásticos

Género: Ciencia ficción. Fantástico. Acción, Superhéroes. Cómic. Marvel Comics

Director: Josh Trank

Puntuación de FilmAffinity: 3.8 sobre 10 de 695 votos

Puntuación sobre tus gustos: No tenemos suficiente información en tu perfil

Fecha de estreno: 21/08/15

Año: 2015

Sinopsis: Cuatro jóvenes inadaptados se teletransportan a un peligroso universo alternativo, lo que les confiere extraños poderes. Cuando sus vidas cambian de forma drástica e irremediable, deben aprender a controlar sus nuevas habilidades y trabajar en equipo para salvar al mundo de un viejo conocido que ahora se ha convertido en un temible enemigo. Adaptación contemporánea del equipo de superhéroes de Marvel. (FILMAFFINITY)

Esta película contiene 37 críticas de usuarios, para oírlas diga oír críticas. Para más detalles de diga más detalles

Tabla 0-62 - Diálogo Ver Película para usuario identificado sin preferencias

Información sobre la película: Cuatro Fantásticos

Título: Cuatro Fantásticos

Género: Ciencia ficción. Fantástico. Acción, Superhéroes. Cómic. Marvel Comics

Director: Josh Trank

Puntuación de FilmAffinity: 3.8 sobre 10 de 695 votos

Puntuación sobre tus gustos: 4 sobre 10.

Fecha de estreno: 21/08/15

Año: 2015

Sinopsis: Cuatro jóvenes inadaptados se teletransportan a un peligroso universo alternativo, lo que les confiere extraños poderes. Cuando sus vidas cambian de forma drástica e irremediable, deben aprender a controlar sus nuevas habilidades y trabajar en equipo para salvar al mundo de un viejo conocido que ahora se ha convertido en un temible enemigo. Adaptación contemporánea del equipo de superhéroes de Marvel. (FILMAFFINITY)

Esta película contiene 37 críticas de usuarios, para oírlas diga oír críticas. Para más detalles de diga más detalles

Tabla 0-63 - Diálogo Ver Película para usuario identificado con preferencias

Si el usuario accede a más detalles de la película se le mostrará una descripción de la película más detallada de la película. Como se puede observar en el siguiente diálogo de ejemplo:

Información Detallada sobre la película: Cuatro Fantásticos

Título: Cuatro Fantásticos

Género: Ciencia ficción. Fantástico. Acción, Superhéroes. Cómic. Marvel Comics

Año: 2015

Director: Josh Trank

Puntuación: 3.8 sobre 10 de 695 votos

Puntuación sobre tus gustos: No tenemos suficiente información en tu perfil

Fecha de estreno: 21/08/15

Reparto: Miles Teller, Kate Mara, Michael B. Jordan, Jamie Bell, Toby Kebbell, Tim Blake Nelson, Reg E. Cathey, Lance E. Nichols, Mary Rachel Dudley, Tim Bell, Ronnie Hooks, Jodi Lyn Brockton, Chet Hanks, Shauna Rappold

Guión: Simon Kinberg, Jeremy Slater (Historia: Josh Trank. Personajes: Stan Lee, Jack Kirby)

País: Estados Unidos

Música: Marco Beltrami, Philip Glass

Fotografía: Matthew Jensen

Producción: 20th Century Fox

Duración: 100 minutos

Críticas: Es estupenda, en los diálogos, el tono, el carisma de sus intérpretes. (...) la acción: tosca, pedestre, efectos especiales de pacotilla. Así que, dependiendo de lo que le pidan a una película de este corte, ya saben lo que aguarda. ¿Un bodrio? No (tanto).

Javier Ocaña: Diario El País Lamentable relanzamiento de franquicia (...) imbuido del síndrome existencialista del superhéroe nolaniano, Trank no ha retenido ni un watio de la electricidad de «Batman Begins» o «El caballero oscuro». Sergi Sánchez: Diario La Razón Aunque no es una baratija que se pueda tirar sin apenas mirarla, tampoco es la joya de la corona. Es un cómic tan celebrado y con tanta riqueza en su interior que resulta extraño (y decepcionante) que tampoco esta vez consigan sacarle brillo. José Manuel Cuéllar: Diario ABC Un guion desastroso, en el que la acción no acaba de arrancar nunca y, cuando parece que va a hacerlo, va y se acaba la película (...) con unos cutres efectos visuales de saldo (...) Puntuación: (sobre 5) Alberto Luchini: Diario El Mundo Es al llegar la hora de las tortas cuando todo se acelera, los nervios de la productora se materializan a machetazos y el clímax vulgariza lo que podría haber sido una bella heterodoxia. (...) Puntuación (sobre 5) Daniel De Partearroyo: Cinemanía

Sinopsis: Cuatro jóvenes inadaptados se teletransportan a un peligroso universo alternativo, lo que les confiere extraños poderes. Cuando sus vidas cambian de forma drástica e irremediable, deben aprender a controlar sus nuevas habilidades y trabajar en equipo para salvar al mundo de un viejo conocido que ahora se ha convertido en un temible enemigo. Adaptación contemporánea del equipo de superhéroes de Marvel. (FILMAFFINITY)

Esta película contiene 37 críticas de usuarios, para oírlas diga oír críticas

Tabla 0-64 - Diálogo Ver película detallada

Una vez en las vistas de descripción de película tanto como en la normal como en la detallada el usuario puede acceder a las críticas de usuario.

Las críticas de usuario serán mostradas paginadas de 1 en 1. Estas críticas pueden tener spoilers de la película por lo que el usuario será avisado en tal caso y se le dará una opción para escuchar la crítica entera con spoilers si lo desea.

Estas críticas serán mostradas de la siguiente manera:

Mostrando Crítica 2 de 305.

Título: He sido una víctima del HYPE

Autor: Armando86

Fecha: 15 de junio de 2015

Puntuación: 4

Crítica: Buenos días a todos.

Escribo esta crítica especialmente para prevenir a cualquiera que como yo, se haya sentido mínimamente emocionado por los comentarios y primeras críticas vertidas sobre esta película. Después del pre-estreno y a través de esta web, leí críticas “de profesionales” en la que se indicaba que esta secuela era la mejor de las tres y que hacía recordar, en algunos momentos, a la opera prima de esta serie que comenzó allá por el '93.

Pues bien, desde mi punto de vista ni es de lejos la mejor de las tres (para mí es la primera secuela, del 97, la mejor de las tres con mucha diferencia) y ni muchísimo menos hace recordar a la original... Miento, la hace recordar para lamentarnos por el engaño por el que acabas de pagar 10€ (sí, 10€, redondos. Cine Proyecciones en la calle Fuencarral de Madrid, un domingo).

¿Por qué me molesto en escribir todo esto?

1. Porque Jurassic Park está entre mis películas favoritas. Hoy, 22 años después, me sigue emocionando.
2. Porque en vez de ir hacia delante vamos hacia detrás. Los dinosaurios son 100% digitales y se ve a la legua que son incrustaciones. Sin ser ningún experto en efectos especiales, los bichos de la película del 93 parecían infinitamente (quisiera recalcar esto) más reales.
3. Las situaciones completamente absurdas que se dan en esta película, la convierten en pasto para ganado de cine palomitero veraniego. Esto lo podría asumir con Transformers o cualquier creación de ese estilo, pero para esta serie mi alma esperaba algo con más profundidad.

Tabla 0-65 - Diálogo Ver críticas de usuario

CAPÍTULO 5. Evaluación de la aplicación

Para verificar el correcto funcionamiento de la aplicación se ha realizado una serie de pruebas sobre la misma. Con estas pruebas se ha perseguido un doble objetivo: estudiar la facilidad de uso de las interfaces Web y de voz mediante la realización de pruebas unitarias, y posteriormente mediante la realización de pruebas de integración permitir eliminar los posibles fallos que hayan aparecido en la aplicación.

Estas pruebas se han dividido en dos tipos de pruebas:

- Pruebas automáticas
- Pruebas manuales

5.1 Pruebas automáticas

Para la fase de pruebas del producto, la infraestructura digital dispone de un Injector para poder realizar pruebas automáticas unitarias y de integración a la aplicación.

El Injector tiene la responsabilidad de devolver instancias de la clase pedida. De esta manera cualquier clase puede ser cambiada por un objeto simulado de esta clase.

El objeto simulado de una clase es una réplica de una instancia a la que se pueden añadir expectativas de llamadas a esta instancia y especificar que debe de devolver en cada llamada. Los objetos simulados también pueden tener predefinido los resultados a devolver para cada método.

De esta manera con el Injector y creando objetos simulados para todas las clases utilizadas en una clases se puede hacer una prueba automática unitaria (la cual solo debe de probar la lógica de la propia clase y no de las clases que utilice dentro de ella).

En cambio para las pruebas automáticas de integración, los cuales han de probar la aplicación entera sin llegar a acceder a servicios de terceros como podría ser la web de FilmAffinity. Se puede cambiar la clase DBConfig para que devuelva la configuración de base de datos local en vez de la configuración de la base de datos real. Además se puede crear un objeto simulado que haga que FilmAffinityApi no haga peticiones a la página FilmAffinity si no que solo las simule.

Para correr las pruebas automáticas se ha hecho uso de la librería de PHP PHPUnit además de una clase de Test padre que carga el entorno de la aplicación, además de especificar la clase padre para las pruebas.

Además se dispone de un fichero con todo el esquema SQL que será utilizado para restaurar la base de datos en cada momento. Con estas pruebas automáticas se puede probar que la lógica de la aplicación funciona correctamente.

5.2 Pruebas manuales y evaluación

Para las pruebas manuales se ha elegido un grupo de 10 personas de distintas edades y sexo las cuales cuáles han ejecutado varios casos de uso de la aplicación

Se han elegido varios casos de uso en el que los usuarios recorran toda la aplicación.

En cada uno de los casos de prueba tras el uso de la aplicación se les ha dado el siguiente cuestionario para poder evaluar su experiencia.

1.- Puntúe en una escala de 1 a 5 el grado en el cuál se ha sentido entendido por el sistema (1=“Muy bajo”, 2=“Bajo”, 3=“Medio”, 4=“Alto”, 5=“Muy alto”).

1 | 2 | 3 | 4 | 5

2.- Puntúe en una escala de 1 a 5 su grado de dificultad a la hora de hacer uso de la aplicación de voz desarrollada (1=“Muy difícil”, 2=“Difícil”, 3=“Normal”, 4=“Fácil”, 5=“Muy fácil”).

1 | 2 | 3 | 4 | 5

3.- Puntúe en una escala de 1 a 5 la utilidad de la información que se presenta a través de la aplicación de voz (1=“No útil”, 2=“Poco útil”, 3=“Útil”, 4=“Bastante útil”, 5=“Muy útil”).

1 | 2 | 3 | 4 | 5

4.- ¿Cree usted que el sistema se comportó de manera similar a como lo haría un humano? (1=“Nunca”, 2=“Casi Nunca”, 3=“A medias”, 4=“Casi siempre”, 5=“Siempre”).

1 | 2 | 3 | 4 | 5

5.- En términos generales, ¿está usted satisfecho con el sistema desarrollado? (1=“No, nada” | 2=“Poco satisfecho” | 3=“Satisfecho” | 4=“Bastante satisfecho” | 5=“Muy satisfecho”)

1 | 2 | 3 | 4 | 5

Los flujos que se han elegido han sido los siguientes:

- Prueba 1: Usuario sin identificar navega por la cartelera y próximos estrenos
- Prueba 2: Usuario sin identificar busca películas por director, actor y por título.
- Prueba 3: Usuario consulta una película y sus críticas
- Prueba 4: Usuario sin registrar se registra y rellena su perfil
- Prueba 5: Usuario identificado consulta las recomendaciones para ir a ver al cine

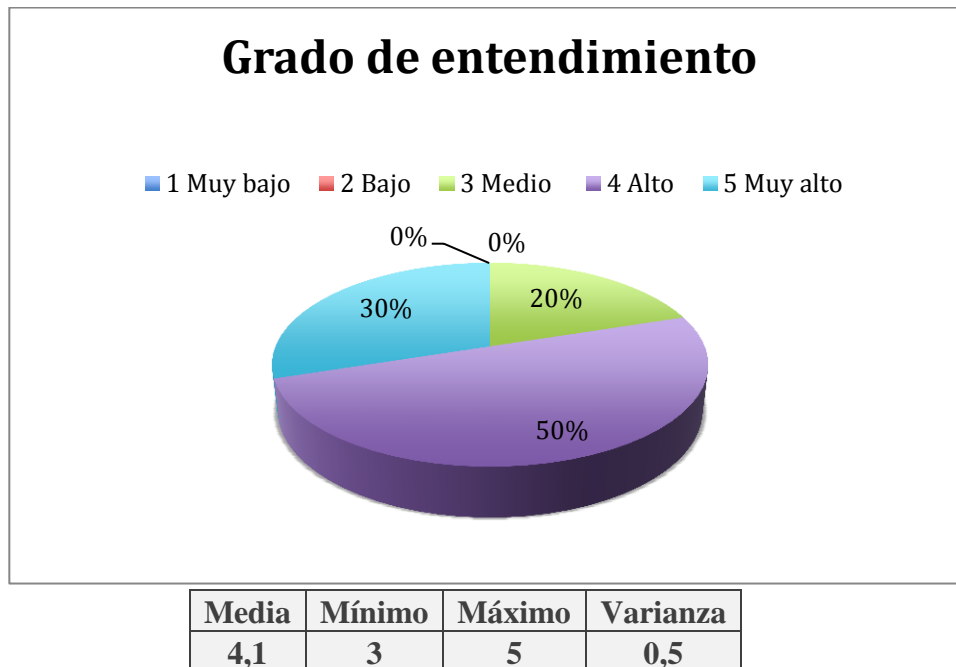
Tras la ejecución de estos flujos se les ha pedido que mencionen que creen que se le podría añadir a la aplicación.

5.2.1 Prueba 1: Usuario sin identificar navega por la cartelera y por los próximos estrenos

En esta prueba el usuario entrara en la aplicación como invitado navegara por la cartelera por sus distintos órdenes y seleccionara una película. Tras esto volverá hacia atrás y navegara por los próximos estrenos

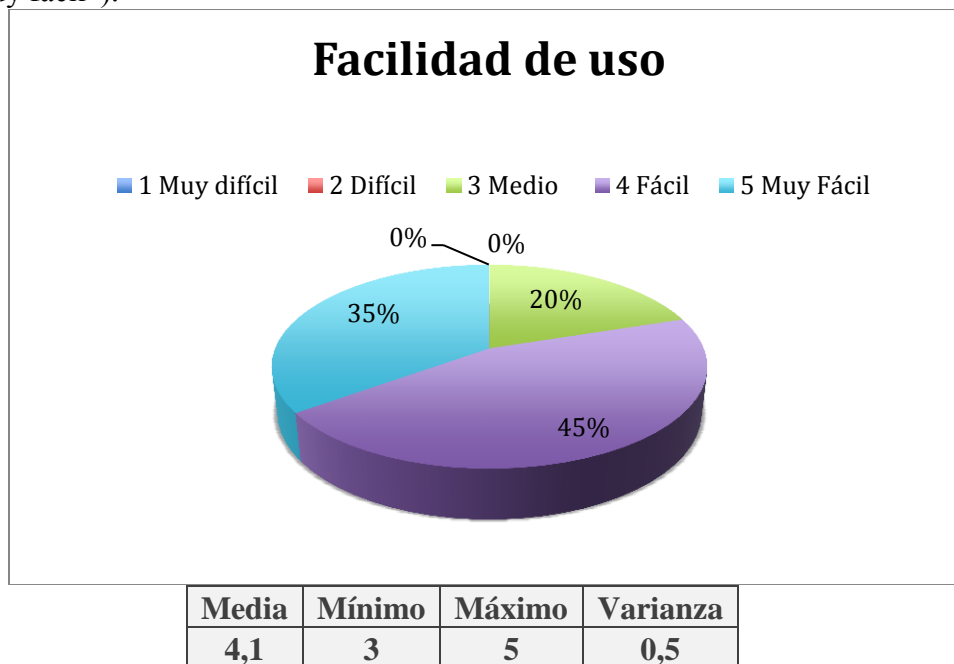
Grado de entendimiento

1.- Puntúe en una escala de 1 a 5 el grado en el cuál se ha sentido entendido por el sistema (1=“Muy bajo”, 2=“Bajo”, 3=“Medio”, 4=“Alto”, 5=“Muy alto”).



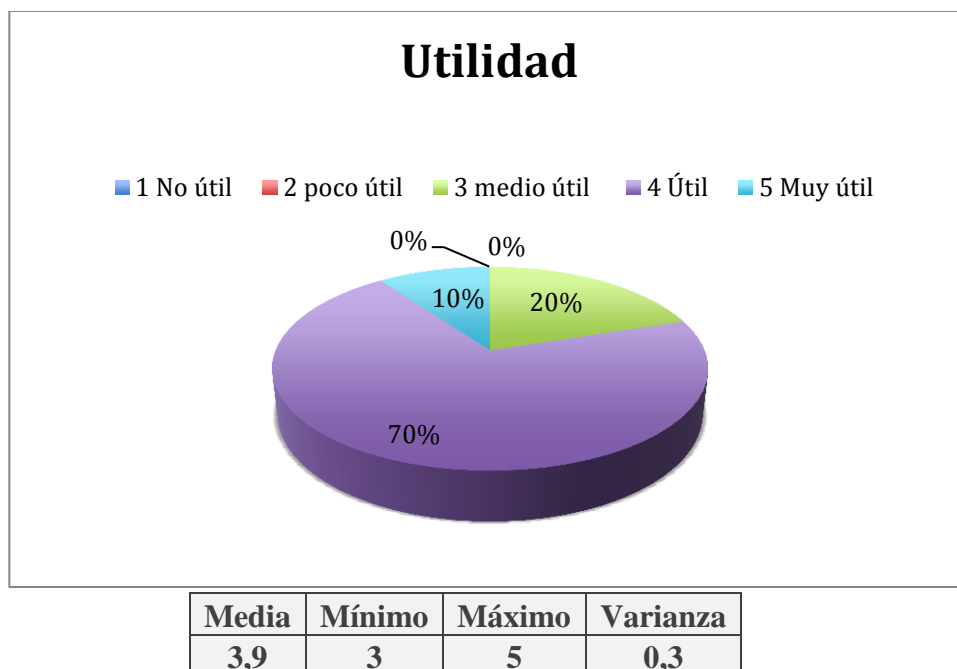
Facilidad de uso

2.- Puntúe en una escala de 1 a 5 su grado de dificultad a la hora de hacer uso de la aplicación de voz desarrollada (1=“Muy difícil”, 2=“Difícil”, 3=“Normal”, 4=“Fácil”, 5=“Muy fácil”).



Utilidad de la aplicación

3.- Puntúe en una escala de 1 a 5 la utilidad de la información que se presenta a través de la aplicación de voz (1=“No útil”, 2=“Poco útil”, 3=“Útil”, 4=“Bastante útil”, 5=“Muy útil”).

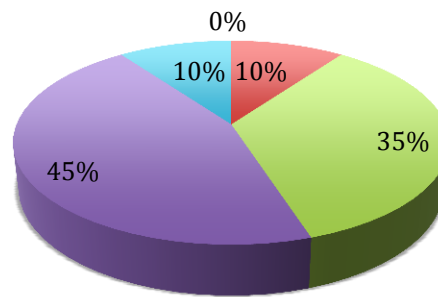


Similitud a un operador humano

4.- ¿Cree usted que el sistema se comportó de manera similar a como lo haría un humano? (1="Nunca", 2="Casi Nunca", 3="A medias", 4="Casi siempre", 5="Siempre").

Similitud a operador humano

■ 1 Nunca ■ 2 Casi nunca ■ 3 A Veces ■ 4 Casi siempre ■ 5 Siempre



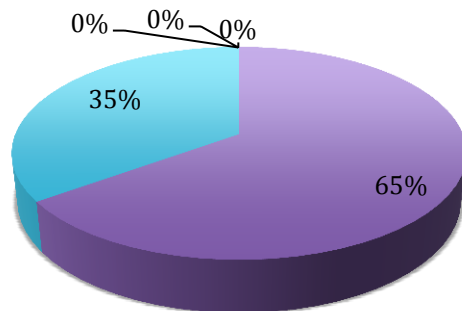
Media	Mínimo	Máximo	Varianza
3,6	2	5	0,7

Satisfacción general

5.- En términos generales, ¿está usted satisfecho con el sistema desarrollado? (1="No, nada" | 2="Poco satisfecho" | 3="Satisfecho" | 4="Bastante satisfecho" | 5="Muy satisfecho").

Satisfacción general

■ 1 No, nada ■ 2 Poco satisfecho ■ 3 Satisfecho
■ 4 Bstante satisfecho ■ 5 Muy satisfecho



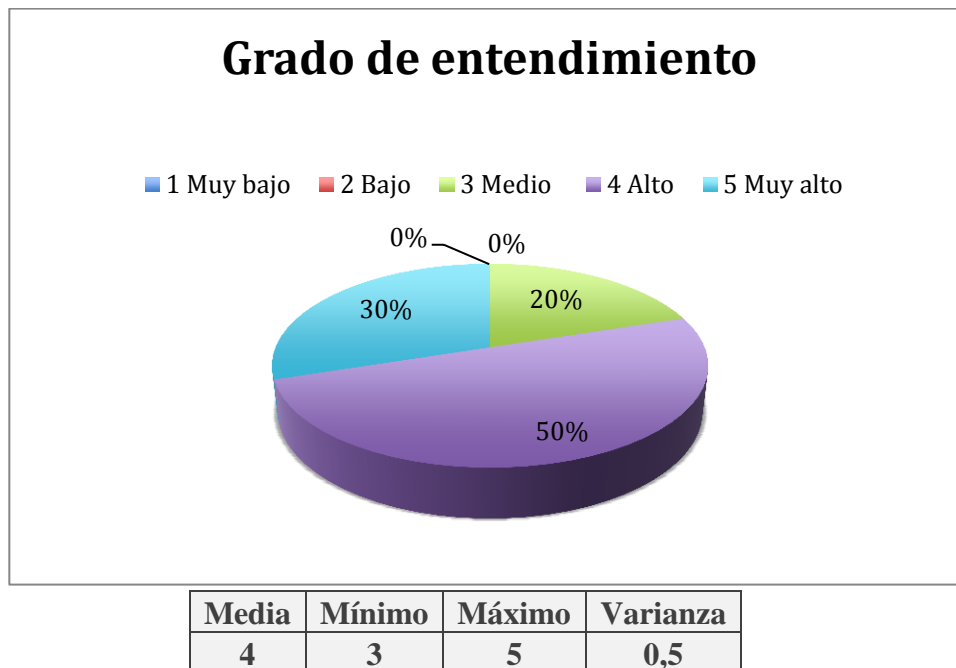
Media	Mínimo	Máximo	Varianza
4,4	4	5	0,2

5.2.2 Prueba 2: Usuario sin identificar busca una película por director, actor y por título.

En esta prueba el usuario entrara en la aplicación como invitado y buscara una película por título, escuchara varias páginas de resultados y elegirá una película.

Grado de entendimiento

1.- Puntúe en una escala de 1 a 5 el grado en el cuál se ha sentido entendido por el sistema (1=“Muy bajo”, 2=“Bajo”, 3=“Medio”, 4=“Alto”, 5=“Muy alto”).

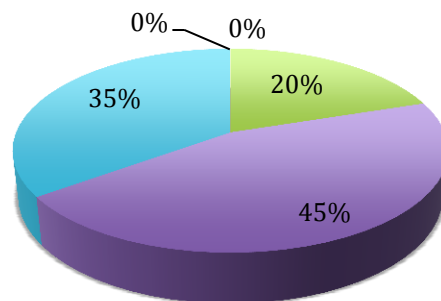


Facilidad de uso

2.- Puntúe en una escala de 1 a 5 su grado de dificultad a la hora de hacer uso de la aplicación de voz desarrollada (1=“Muy difícil”, 2=“Difícil”, 3=“Normal”, 4=“Fácil”, 5=“Muy fácil”).

Facilidad de uso

■ 1 Muy difícil ■ 2 Difícil ■ 3 Medio ■ 4 Fácil ■ 5 Muy Fácil



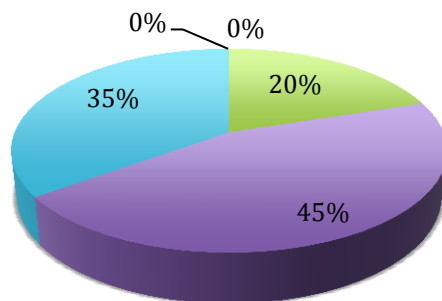
Media	Mínimo	Máximo	Varianza
4,3	3	5	0,4

Utilidad de la aplicación

3.- Puntúe en una escala de 1 a 5 la utilidad de la información que se presenta a través de la aplicación de voz (1=“No útil”, 2=“Poco útil”, 3=“Útil”, 4=“Bastante útil”, 5=“Muy útil”).

Utilidad

■ 1 Muy difícil ■ 2 Difícil ■ 3 Medio ■ 4 Fácil ■ 5 Muy Fácil



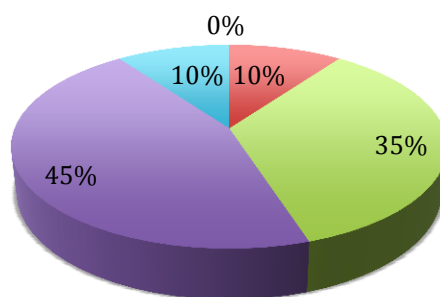
Media	Mínimo	Máximo	Varianza
4	3	5	0,3

Similitud a un operador humano

4.- ¿Cree usted que el sistema se comportó de manera similar a como lo haría un humano? (1="Nunca", 2="Casi Nunca", 3="A medias", 4="Casi siempre", 5="Siempre").

Similitud a operador humano

■ 1 Nunca ■ 2 Casi nunca ■ 3 A Veces ■ 4 Casi siempre ■ 5 Siempre



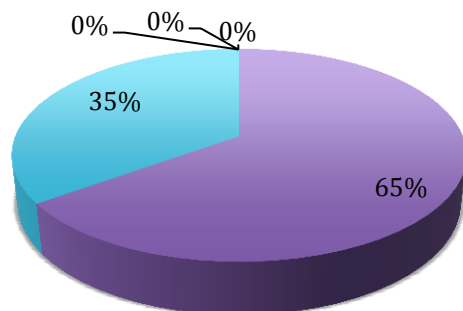
Media	Mínimo	Máximo	Varianza
3,7	2	5	0,7

Satisfacción general

5.- En términos generales, ¿está usted satisfecho con el sistema desarrollado? (1="No, nada" | 2="Poco satisfecho" | 3="Satisfecho" | 4="Bastante satisfecho" | 5="Muy satisfecho")

Satisfacción general

■ 1 No, nada ■ 2 Poco satisfecho ■ 3 Satisfecho
■ 4 Bstante satisfecho ■ 5 Muy satisfecho



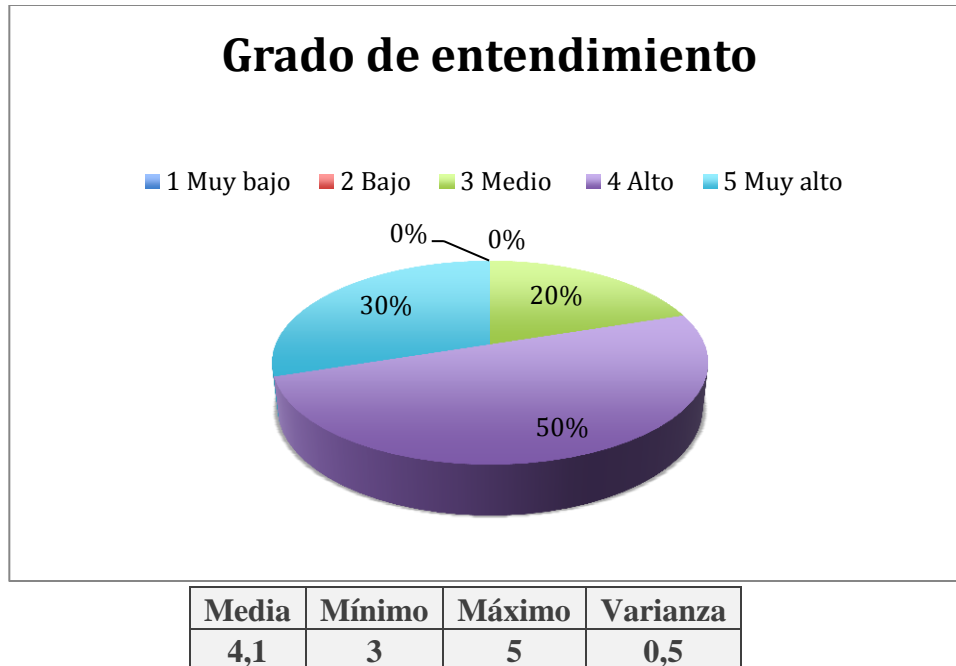
Media	Mínimo	Máximo	Varianza
4,5	4	5	0,3

5.2.3 Prueba 3: Usuario consulta una película y sus críticas

En esta prueba accederán a la información de una película como ellos quieran y accederán a su contenido así como a sus críticas.

Grado de entendimiento

1.- Puntúe en una escala de 1 a 5 el grado en el cuál se ha sentido entendido por el sistema (1=“Muy bajo”, 2=“Bajo”, 3=“Medio”, 4=“Alto”, 5=“Muy alto”).

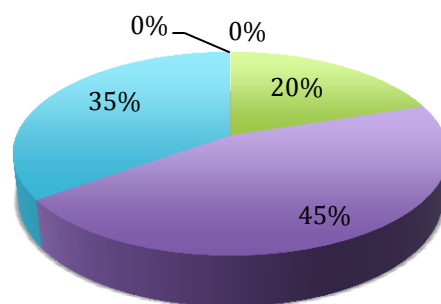


Facilidad de uso

2.- Puntúe en una escala de 1 a 5 su grado de dificultad a la hora de hacer uso de la aplicación de voz desarrollada (1=“Muy difícil”, 2=“Difícil”, 3=“Normal”, 4=“Fácil”, 5=“Muy fácil”).

Facilidad de uso

■ 1 Muy difícil ■ 2 Difícil ■ 3 Medio ■ 4 Fácil ■ 5 Muy Fácil



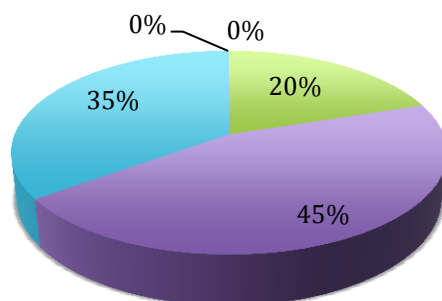
Media	Mínimo	Máximo	Varianza
4,4	3	5	0,3

Utilidad de la aplicación

3.- Puntúe en una escala de 1 a 5 la utilidad de la información que se presenta a través de la aplicación de voz (1=“No útil”, 2=“Poco útil”, 3=“Útil”, 4=“Bastante útil”, 5=“Muy útil”).

Utilidad

■ 1 Muy difícil ■ 2 Difícil ■ 3 Medio ■ 4 Fácil ■ 5 Muy Fácil



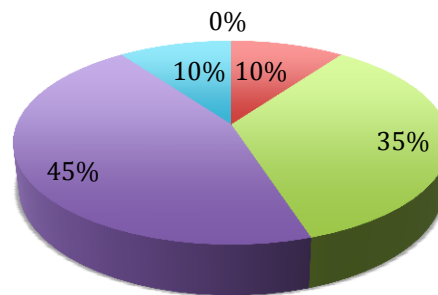
Media	Mínimo	Máximo	Varianza
4,1	3	5	0,3

Similitud a un operador humano

4.- ¿Cree usted que el sistema se comportó de manera similar a como lo haría un humano? (1="Nunca", 2="Casi Nunca", 3="A medias", 4="Casi siempre", 5="Siempre").

Similitud a operador humano

■ 1 Nunca ■ 2 Casi nunca ■ 3 A Veces ■ 4 Casi siempre ■ 5 Siempre



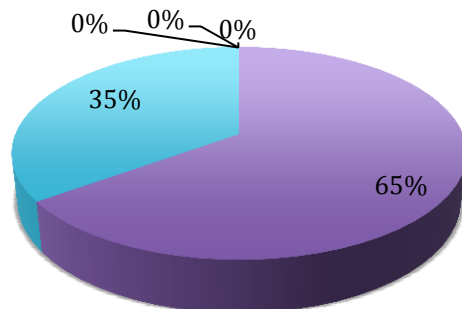
Media	Mínimo	Máximo	Varianza
3,6	2	5	0,7

Satisfacción general

5.- En términos generales, ¿está usted satisfecho con el sistema desarrollado? (1="No, nada" | 2="Poco satisfecho" | 3="Satisfecho" | 4="Bastante satisfecho" | 5="Muy satisfecho")

Satisfacción general

■ 1 No, nada ■ 2 Poco satisfecho ■ 3 Satisfecho
■ 4 Bstante satisfecho ■ 5 Muy satisfecho



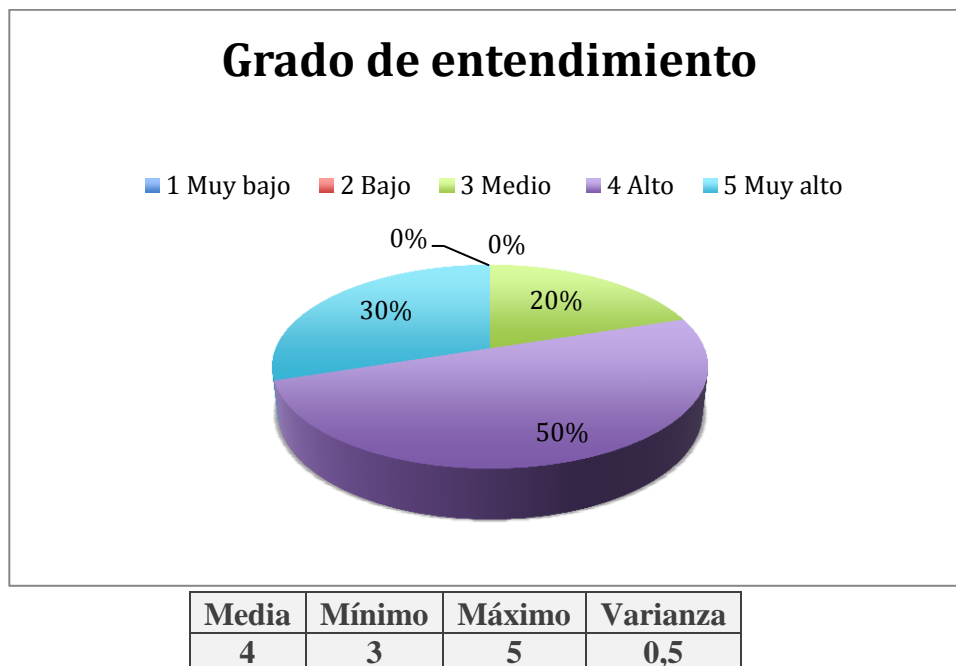
Media	Mínimo	Máximo	Varianza
4,5	4	5	0,3

5.2.4 Prueba 4: Usuario sin registrar se registra y rellena su perfil

En esta prueba los usuarios se registraran, accederán a su perfil y rellenaran cada uno de los directores, actores, géneros, cines y horarios favoritos. También rellenaran la lista negra de actores, géneros y directores.

Grado de entendimiento

1.- Puntúe en una escala de 1 a 5 el grado en el cuál se ha sentido entendido por el sistema (1=“Muy bajo”, 2=“Bajo”, 3=“Medio”, 4=“Alto”, 5=“Muy alto”).

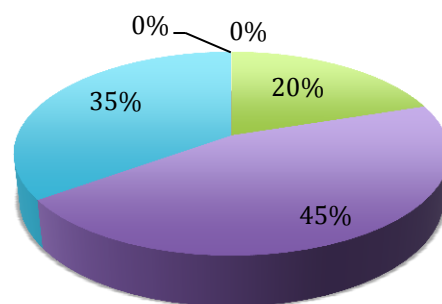


Facilidad de uso

2.- Puntúe en una escala de 1 a 5 su grado de dificultad a la hora de hacer uso de la aplicación de voz desarrollada (1=“Muy difícil”, 2=“Difícil”, 3=“Normal”, 4=“Fácil”, 5=“Muy fácil”).

Facilidad de uso

■ 1 Muy difícil ■ 2 Difícil ■ 3 Medio ■ 4 Fácil ■ 5 Muy Fácil



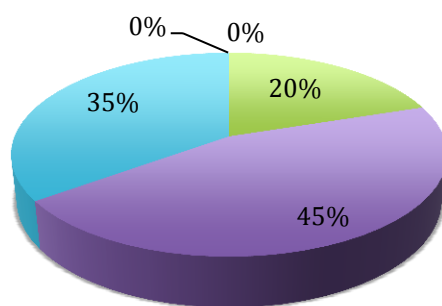
Media	Mínimo	Máximo	Varianza
4,3	3	5	0,4

Utilidad de la aplicación

3.- Puntúe en una escala de 1 a 5 la utilidad de la información que se presenta a través de la aplicación de voz (1="No útil", 2="Poco útil", 3="Útil", 4="Bastante útil", 5="Muy útil").

Utilidad

■ 1 Muy difícil ■ 2 Difícil ■ 3 Medio ■ 4 Fácil ■ 5 Muy Fácil



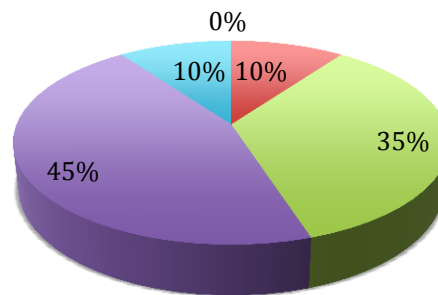
Media	Mínimo	Máximo	Varianza
4	3	5	0,3

Similitud a un operador humano

4.- ¿Cree usted que el sistema se comportó de manera similar a como lo haría un humano? (1="Nunca", 2="Casi Nunca", 3="A medias", 4="Casi siempre", 5="Siempre").

Similitud a operador humano

■ 1 Nunca ■ 2 Casi nunca ■ 3 A Veces ■ 4 Casi siempre ■ 5 Siempre



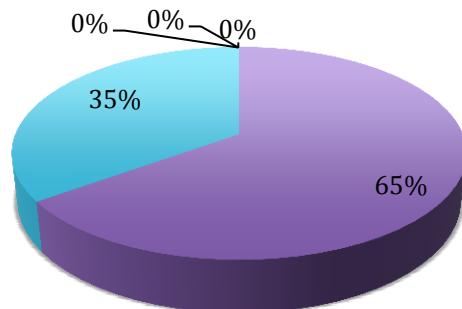
Media	Mínimo	Máximo	Varianza
3,7	2	5	0,7

Satisfacción general

5.- En términos generales, ¿está usted satisfecho con el sistema desarrollado? (1="No, nada" | 2="Poco satisfecho" | 3="Satisfecho" | 4="Bastante satisfecho" | 5="Muy satisfecho")

Satisfacción general

■ 1 No, nada ■ 2 Poco satisfecho ■ 3 Satisfecho
■ 4 Bstante satisfecho ■ 5 Muy satisfecho



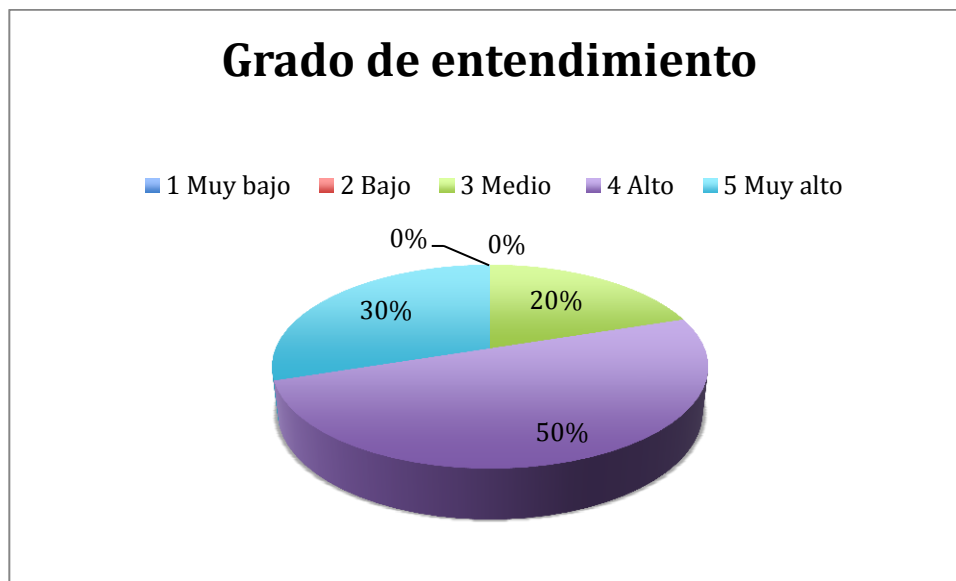
Media	Mínimo	Máximo	Varianza
4,5	4	5	0,3

5.2.5 Prueba 5: Usuario identificado consulta las recomendaciones para ir a ver al cine

En esta prueba el usuario entrara en la aplicación como invitado y buscara una película por título, escuchara varias páginas de resultados y elegirá una película.

Grado de entendimiento

1.- Puntúe en una escala de 1 a 5 el grado en el cuál se ha sentido entendido por el sistema (1=“Muy bajo”, 2=“Bajo”, 3=“Medio”, 4=“Alto”, 5=“Muy alto”).



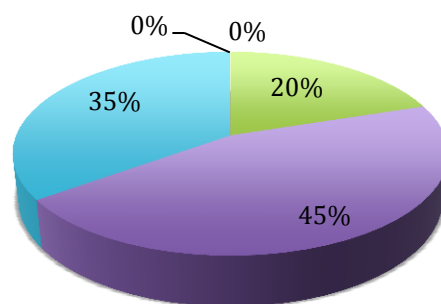
Media	Mínimo	Máximo	Varianza
4,2	3	5	0,5

Facilidad de uso

2.- Puntúe en una escala de 1 a 5 su grado de dificultad a la hora de hacer uso de la aplicación de voz desarrollada (1=“Muy difícil”, 2=“Difícil”, 3=“Normal”, 4=“Fácil”, 5=“Muy fácil”).

Facilidad de uso

■ 1 Muy difícil ■ 2 Difícil ■ 3 Medio ■ 4 Fácil ■ 5 Muy Fácil



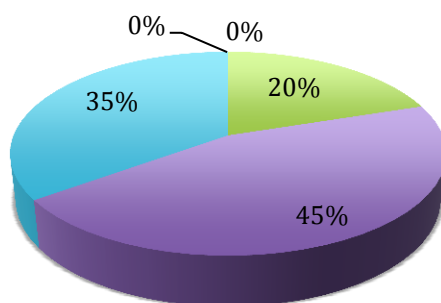
Media	Mínimo	Máximo	Varianza
4,5	3	5	0,4

Utilidad de la aplicación

3.- Puntúe en una escala de 1 a 5 la utilidad de la información que se presenta a través de la aplicación de voz (1="No útil", 2="Poco útil", 3="Útil", 4="Bastante útil", 5="Muy útil").

Utilidad

■ 1 Muy difícil ■ 2 Difícil ■ 3 Medio ■ 4 Fácil ■ 5 Muy Fácil



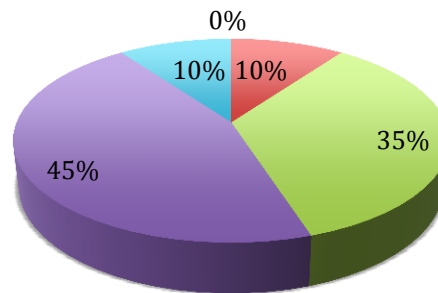
Media	Mínimo	Máximo	Varianza
4,2	3	5	0,3

Similitud a un operador humano

4.- ¿Cree usted que el sistema se comportó de manera similar a como lo haría un humano? (1="Nunca", 2="Casi Nunca", 3="A medias", 4="Casi siempre", 5="Siempre").

Similitud a operador humano

■ 1 Nunca ■ 2 Casi nunca ■ 3 A Veces ■ 4 Casi siempre ■ 5 Siempre



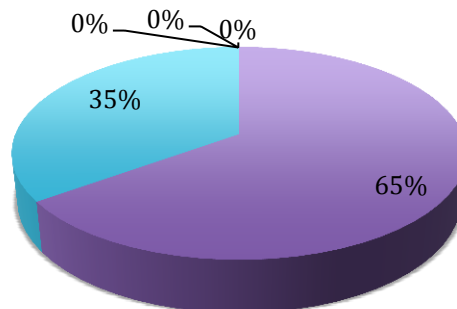
Media	Mínimo	Máximo	Varianza
3,6	2	5	0,6

Satisfacción general

5.- En términos generales, ¿está usted satisfecho con el sistema desarrollado? (1="No, nada" | 2="Poco satisfecho" | 3="Satisfecho" | 4="Bastante satisfecho" | 5="Muy satisfecho")

Satisfacción general

■ 1 No, nada ■ 2 Poco satisfecho ■ 3 Satisfecho
■ 4 Bstante satisfecho ■ 5 Muy satisfecho



Media	Mínimo	Máximo	Varianza
4,5	4	5	0,3

5.2.6 Conclusiones de las pruebas manuales

Analizando los resultados obtenidos en las pruebas anteriores podemos deducir lo siguiente:

- El registro en la aplicación en la navegación por lo general les ha parecido sencilla y útil.
- Por lo general el sistema les parece fácilmente usable a la mayoría de los usuarios.
- El reconocimiento de voz en menús cerrados es más sencillo para los usuarios y se sienten mejor entendidos que en los formularios abiertos (búsquedas). Esto parece ser debido a que de momento la aplicación está configurada con un número reducido de nombres de actores, directores y títulos de películas.
- Los usuarios notan que están hablando con una máquina y no les parece que estén hablando con un operador humano, sobre todo en los menús cerrados por lo que se podría revisar los guiones de los menús para hacerlos menos coloquiales. Pero en todo caso este será un punto para mejorar de los futuros.
- Entre las opiniones sobre qué añadir a la aplicación hay dos opiniones repetidas:
 - Poder comprar entradas en la aplicación.
 - Poder ver la cartelera de un cine sin estar identificado.
- Las dos ideas son muy buenas ideas y la segunda es fácilmente realizable, por lo que podría ser lo primero para hacer como trabajo futuro. La primera en cambio es un poco más complicada por el manejo de tarjetas de crédito en la aplicación, pero también se podría plantear para estudiar implementarla.

CAPÍTULO 6. Conclusiones y trabajo futuro

En este apartado veremos las conclusiones a las que se han llegado con la realización de este proyecto final de carrera y en las mejoras y extensiones que se podrían hacer a esta aplicación en un futuro.

6.1 Conclusiones

Como hemos explicado al principio de la memoria, con esta aplicación lo que se ha intentado es crear una infraestructura digital que ayudase a cualquier desarrollador con poca experiencia o sin conocimientos de VXML a poder desarrollar aplicaciones en VXML con rapidez y sin tener que escribir apenas VXML.

Para ello se ha creado la infraestructura digital y una prueba de concepto que ha sido la aplicación FilmAffinityVxml que es una forma de acceder al contenido de la página FilmAffinity de manera sencilla y eficaz.

Las aplicaciones en VXML tienen varias ventajas respecto a las aplicaciones tradicionales, las aplicaciones VXML son habladas por lo que el usuario solo necesita escuchar la información y el manejo de ellas puede ser totalmente por voz y en algunos casos también se puede manejar mediante el teclado telefónico. Esto hace que esta aplicación pueda ser fácilmente manejada andando por la calle o incluso en el coche con manos libres, mientras que una app no. Además, el manejo de una aplicación de VXML tiene una gran accesibilidad para usuarios con discapacidades visuales o motoras.

Para realizar la infraestructura digital se ha intentado hacer lo más sencillo posible para que una vez implementada fuese sencillo de probar, por ello, toda la infraestructura digital está hecha con formularios VXML que realizan peticiones de tipo HTTP POSTS. De esta manera no tenemos extensas plantillas de VXML, que para un usuario con pocos conocimientos de VXML le sería más complicado de entender y realizar.

Una vez realizado la infraestructura digital y la prueba de concepto con FilmAffinity nos hemos dado cuenta de lo fácil que es crear una aplicación nueva. De hecho en mi trabajo pienso proponer crear una aplicación propia para nuestros usuarios.

Terminado este proyecto, puedo concluir que me ha ayudado mucho para aprender mucho sobre VXML, así como hacer técnicas de web scripting y ampliar mis conocimientos sobre PHP y Mysql. Además de conocimientos sobre hosting, curl options, y configuraciones de hosts en las cuales en mi trabajo diario no suelo usar y me han resultado muy interesantes.

Además, algunas de las asignaturas cursadas durante la carrera (como, por ejemplo, Diseño de Bases de Datos, Gestión de Proyectos, Ingeniería del Software o Sistemas Hipermedia) me han sido de gran utilidad durante las distintas fases del proyecto.

Gracias a la realización de este proyecto he podido darme cuenta de todo lo que conlleva el desarrollo de cualquier aplicación o producto software, desde su diseño y planificación hasta su desarrollo, implementación y evaluación, y cómo van apareciendo situaciones inesperadas que debemos resolver sobre la marcha.

Por último, me gustaría destacar que he disfrutado mucho implementando la aplicación ya que al haberla hecho poco a poco he visto como ha crecido, como he encontrado problemas inesperados y como los hemos sorteado. Todo esto me ha hecho sentir muy orgulloso una vez que la aplicación ha estado completa.

6.2 Trabajo futuro.

Para las líneas futuras de trabajo hemos dividido las líneas de trabajo en dos grupos:

- Trabajo futuro para mejorar la infraestructura digital.
- Trabajo futuro para mejorar FilmAffinityVxml.

6.2.1 Trabajo futuro para mejorar la infraestructura digital

Sistema multilinguaje

Con esta mejora se daría una infraestructura digital totalmente multilinguaje.

Ahora mismo la infraestructura digital es capaz de recibir respuestas del usuario en distintos idiomas, pero no es capaz de devolver la información al usuario en distintos idiomas.

Para ello se necesitaría crear nuevos Templates de manera que se tuviese un Template por acción e idioma, esto se haría así y no con un sistema de traducciones porque la estructura del Template podría cambiar entre idiomas.

Gramáticas generadas automáticamente dependiendo de los datos obtenidos.

Actualmente para elegir una película se ha de decir un número para elegirla ya que el título de la película es obtenido por un sistema externo en la cual podría tener un texto que no fuese valido como un posible input de la gramática para reconocer la opción deseada por el usuario.

Esta mejora lo que haría sería que el sistema podría recoger la información del sistema externo y lo transformaría en una frase reconocible válida, para ello deberíamos de crear

un sistema más completo de limpieza de gramáticas que dada una opción reconocible te diese la opción más parecida que pueda ser reconocida por una gramática.

6.2.2 Trabajo futuro para mejorar FilmAffinityVxml

Estas mejoras han sido recogidas de las propuestas recogidas en la fase de pruebas y las cuales no se han añadido por falta de prueba.

Compra de entradas desde la aplicación

En esta mejora se añadiría un módulo en el que una vez que el usuario está consultando las proyecciones disponibles para una película este podría acceder a comprar las entradas.

Añadir un módulo de compra de entradas para los usuarios supondría tener que poder transmitir números de tarjeta por nuestro servidor, para ello necesitaríamos tener un servidor seguro con certificado para poder almacenar tarjeta. Además deberíamos de comprobar si el sistema de compras o pagos puede ser accedido desde otro servidor o si al menos tiene algún sistema VXML al que conectarse para poder realizar la compra.

Si el cine tuviese alguna manera de hacer la compra desde servidor se debería de diseñar un guion que le explicara al usuario sencillamente que sitios libres hay en una sala y que le fuese fácil la elección de butaca.

Consulta de cartelera de cines para usuarios anónimos

En esta mejora se añadiría un módulo que recomendase películas sin tener en cuenta preferencias y que solo consultase el cine elegido por el usuario en vez de esperar a que el usuario se registre y guarde sus cines favoritos.

Esta mejora es bastante sencilla ya que sólo haría falta tiempo de desarrollo ya que la infraestructura digital permite perfectamente esta opción.

Esta mejora no fue añadida inicialmente porque se quería incentivar que el usuario se registrase y guardase sus preferencias para poder plantear proyectos favoritos.

Sistema de SMS con notificación sobre nuevas películas.

En esta mejora se añadiría una infraestructura digital de envío de SMS que mandarían notificaciones al usuario para notificarle sobre nuevas películas, esto podría ser definido por el usuario para que le avisase sobre estrenos de películas de sus directores, actores o géneros favoritos.

Para crear esta mejora deberíamos de cambiar la forma de guardar las preferencias del usuario para poder buscar por sus preferencias. Esto supondría crear nuevas tablas para guardar todos los directores, actores y géneros guardados por el usuario.

Además se debería de añadir un sistema de envío de SMS por lo que se debería de contratar los servicios de alguna proveedora TELCO para ello.

Tras tener todo lo anterior se debería de crear CRONS con scripts que buscase usuarios a los que enviar las notificaciones y enviárselas periódicamente.

Humanización de los diálogos de la aplicación.

En esta mejora se intentará hacer más humano el diálogo de manera que la gente no se piense que está hablando con una maquina en todo momento.

Para ello se intentará evitar repetir frases en la conversación e introducir varias formas de decir un menú de esta manera si cada vez que hables con el sistema te lo dice distinto te dará mayor sensación de humanidad.

Además se intentará bajar el nivel de trato de usted a tu de la aplicación para sentir darle al usuario una sensación de cercanía y de amigo.

GLOSARIO

- **Sistema de diálogo:** son programas informáticos cuya finalidad es interactuar con los usuarios oralmente o de forma multimodal para proporcionar diversos servicios.
- **URI (Uniform Resource Identifier):** es una cadena corta de caracteres que identifica inequívocamente un recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia, etc.). Normalmente estos recursos son accesibles en una red o sistema. Algunas URI pueden ser URL, URN o ambos.
- **VoiceXML (Voice eXtensible Markup Language):** es un formato XML estándar del W3C para la especificación de diálogos interactivos entre un humano y una máquina.
- **Voxeo:** es una plataforma de implementación del estándar VoiceXML que cumple por completo con las especificaciones del W3C para VoiceXML 2.0 y 2.1.
- **W3C (Consortio World Wide Web):** es una asociación internacional formada por organizaciones miembros del consorcio, personal y el público en general, que trabajan conjuntamente para desarrollar estándares web. W3C pretende guiar la web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la web.
- **XML (eXtensible Markup Language):** es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.
- **MVC:** El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.^{1 2} Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

- **Objeto Simulado:** En la programación orientada a objetos se llaman objetos simulados (pseudoobjetos, mock object, objetos de pega) a los objetos que imitan el comportamiento de objetos reales de una forma controlada. Se usan para probar a otros objetos en pruebas unitarias que esperan mensajes de una clase en particular para sus métodos, al igual que los diseñadores de autos usan un crash dummy cuando simulan un accidente. En los test de unidad, los objetos simulados se usan para simular el comportamiento de objetos complejos cuando es imposible o impracticable usar al objeto real en la prueba. De paso resuelve el problema del caso de objetos interdependientes, que para probar el primero debe ser usado un objeto no probado aún, lo que invalida la prueba: los objetos simulados son muy simples de construir y devuelven un resultado determinado y de implementación directa, independientemente de los complejos procesos o interacciones que el objeto real pueda tener.
- **Prueba automática unitaria:** En programación, una prueba unitaria es una forma de comprobar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Luego, con las Pruebas de Integración, se podrá asegurar el correcto funcionamiento del sistema o subsistema en cuestión.
- **Prueba automática de integración:** Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez.

Consiste en realizar pruebas para verificar que un gran conjunto de partes de software funcionan juntos. Las pruebas de integración (algunas veces llamadas integración y testeo I&T) es la fase de la prueba de software en la cual módulos individuales de software son combinados y probados como un grupo. Son las pruebas posteriores a las pruebas unitarias y preceden a las pruebas del sistema.

BIBLIOGRAFÍA

- [1] López-Cózar, R., and Araki, M.: „Spoken, Multilingual and Multimodal Dialogue Systems: Development and Assessment“ (Wiley, 2005).
- [2] Griol, David. Desarrollo y evaluación de Diferentes Metodologías para la Gestión Automática del Diálogo. Tesis Doctoral. Universidad Politécnica de Valencia, 2007.
- [3] Llisterri, J. (2006) "Introducción a los sistemas de diálogo", in LLISTERRI, J.-MACHUCA, M. J. (Eds.) Los sistemas de diálogo. Bellaterra - Soria: Universitat Autònoma de Barcelona, Servei de Publicacions - Fundación Duques de Soria (Manuals de la Universitat Autònoma de Barcelona, Lingüística, 45), págs. 11- 21.
- [4] Jorge Hierro Álvarez. Informe técnico sobre los sistemas de reconocimiento de voz. Departamento de Prensa Assit. Madrid, Junio de 2004. Disponible en: <http://jorgehierro.files.wordpress.com/2008/02/voice-reconigtionii.pdf>
- [5] B. Green, A. Wolf, C. Chomsky, K. Laughery. BASEBALL: An Automatic Question .Proc.of the Western Joint Computer Conference. 1963, páginas 219- 224.
- [6] D.G. Bobrow. Natural Language Input for a Computer Problem-solving System. M. Minsky B.(ed.) Semantic Information Processing. Cambridge, Mass: MIT Press.1968, páginas 146-226.
- [7] T. Winograd. Understanding Natural Language. Academic Press Eds. 1972.
- [8] Joseph Weizenbaum. ELIZA, A Computer Program for the Study of Natural Language Communication between Man and Machine. Communications of the ACM. 1965, volumen 9, páginas 36-45.
- [9] Kenneth Mark Colby. Artificial Paranoia: A Computer Simulation of Paranoid Processes. Elsevier Science Inc. New York, NY. 1975.
- [10] Minker, W., Haiber, U., Heisterkamp, P., Scheible, S. “The Seneca Spoken Language Dialogue System”. Speech Communication 43, 2004.
- [11] Weng, F., Varges, S., Raghunathan, B., Ratiu, F., Pon-Barry, H., Lathrop, B., Zhang, Q., Scheideck, T., Bratt, H., Xu, K., Purver, M., Mishra, R., Raya, M., Peters, S., Meng, Y., Cavedon, L., Shriberg, L. “CHAT: A Conversational Helper for Automotive Tasks”. En: Proc. of the 9th International Conference on Spoken Language Processing (Interspeech’06-ICSLP). 2006. (pg. 1061-1064).

- [12] Andreani, G., Di Fabrizio, G., Gilbert, M., Gillick, D., Hakkani-Tur, D., Lemon, O. "Let's DISCOH: Collecting an Annotated Open Corpus with Dialogue Acts and Reward Signals for Natural Language Helpdesks." En: Proc. of IEEE 2006 Workshop on Spoken Language Technology (SLT'06). 2006. (pg. 218- 221).
- [13] Bohus, D., Grau, S., Huggins-Daines, D., Keri, V., Krishna, G., Kumar, R., Raux, A., Tomko, S. "Conquest - an Open-Source Dialog System for Conferences." En: Proc. of Human Language Technologies'07: The Conference of the North American Chapter of the Association for Computational Linguistics. 2007. (pg. 9-12).
- [14] Roque, A., Leuski, A., Rangarajan, V., Robinson, S., Vaswani, A., Narayanan, S., Traum, D., "Radiobot-CFF: A Spoken Dialogue System for Military Training." En: Proc. of the 9th International Conference on spoken Language Processing (Interspeech'06-ICSLP). 2006. (pg. 477-480).
- [15] Lluís Hurtado, Fernando Blat, Sergio Grau, David Griol, Emilio Sanchis, Encarna Segarra, Francisco Torres. Sistema de diálogo para el Proyecto DIHANA. Procesamiento del Lenguaje Natural, ISSN 1135-5948, No. 35, 2005, págs. 453-454.
- [16] " Uniform Resource Identifiers (URI): Generic Syntax ", IETF RFC 2396, 1998. Disponible en: <http://www.ietf.org/rfc/rfc2396.txt> [septiembre de 2015]
- [17] Voxeo Evolution. Disponible en: <http://www.evolution.voxeo.com> [septiembre de 2015]
- [18] Speech Recognition Grammar Specification Version 1.0. <http://www.w3.org/TR/speech-grammar/> [septiembre de 2015]
- [19] Semantic Interpretation for Speech Recognition (SISR) Version 1.0 <http://www.w3.org/TR/semantic-interpretation/> [septiembre de 2015]
- [20] Batacharia B., et al. "CONVERSE: a Conversational Companion Machine Conversations". En: Proc. of the 1st International Workshop on Human-Computer Conversation. 1999.